(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 226 353 B1**

(12) # EUROPEAN PATENT SPECIFICATION

(45) Date of publication of patent specification: **07.04.93** (51) Int. Cl.5: **H03M 13/00**, //G06F11/10

(21) Application number: **86309176.5**

(22) Date of filing: **25.11.86**

(54) **CRC calculation apparatus.**

(30) Priority: **02.12.85 US 803351**

(43) Date of publication of application:
24.06.87 Bulletin 87/26

(45) Publication of the grant of the patent:
07.04.93 Bulletin 93/14

(84) Designated Contracting States:
AT BE CH DE ES FR GB GR IT LI LU NL SE

(56) References cited:
US-A- 3 678 469
US-A- 4 171 765

COMPUTER DESIGN. vol. 11, no. 5, May 1972, LITTLETON, MASSACHUS pages 95 - 98; PAUL KRAEGER: "SMALL COMPUTER CYCLIC REDUNDANCY CHECKSUM CAPABILITY"

COMPUTER DESIGN. vol. 14, no. 9, September 1975, LITTLETON, MASSACHUS pages 87 - 91; A.K.PANDEYA ET AL: "PARALLEL CRC LETS MANY LINES USE ONE CIRCUIT"

MESURES REGULATION AUTOMATISME. vol. 49, no. 7, May 1984, PARIS FR pages 43 - 46; J.P.VERNAY: "TRANSMISSION DE DONNEES: GAGNEZ DE LA PLACE"

(73) Proprietor: **ADVANCED MICRO DEVICES, INC.**
901 Thompson Place P.O. Box 3453
Sunnyvale, CA 94088(US)

(72) Inventor: **Joshi, Sunil P.**
4844 Westmont Avenue
Campbell California 95008(US)
Inventor: **Iyer, Venkatraman**
2600 Buena Vista Way
Berkeley California 94708(US)

(74) Representative: **Sanders, Peter Colin Christopher et al**
BROOKES & MARTIN High Holborn House
52/54 High Holborn
London WC1V 6SE (GB)

EP 0 226 353 B1

Description

The invention pertains to the field of cyclical redundancy codes calculation circuits for detecting errors in transmission of serial data over communication links. More particularly, the invention pertains to certain improvements in CRC calculation apparatus in integrated circuit form to conserve chip area.

Error detection and correction using codes and check bits has long been used to improve the reliability of data transfers between units on a network or units in a computer system such as disk or main memory and the central processing unit. The most common scheme is parity checking. In this scheme, parity check bits are added to the information bits to make the total number of bits which are logic 1's in a byte equal to a known number. However this scheme has the well known drawback that when the number of information bits becomes high, the level of redundancy in terms of check bits required becomes excessively high.

Another checking scheme exists called polynomial or cyclic coding. This scheme can be designed to perform with higher efficiencies, i.e., less redundancy, than the parity checking schemes. The high efficiency of these schemes is inducing designers to use them more and more frequently.

The general concepts of cyclic coding schemes are most easily understood through use of several mental aids. A convenient way of thinking of a bit stream of data in serial format consisting of K bits is to think of it as a polynomial in a dummy variable x with K terms. The bits of the message are the coefficients of the polynomial. Thus, if 100100011011 is the bit stream message, the polynomial may be written as:

$$(1) \quad M(x) = 1.x^{11} + 0.x^{10} + 0.x^{9} + 1.x^{8} + 0.x^{7} + 0.x^{6} + 0.x^{5} + 1.x^{4} + 1.x^{3} + 0.x^{2} + 1.x^{1} + 1.x^{0}$$

or

$$M(x) = x^{11} + x^{8} + x^{4} + x^{3} + x + 1$$

To compute the cyclic code check bits (hereafter CRC bits) on a message, another polynomial P(x) called a generating polynomial is chosen. The degree of this polynomial, i.e., its highest exponent value, is greater than zero but less than the degree of M(x). The generator polynomial has a non-zero coefficient in the x^0 term. For a message of a given length, more than one generating polynomial can be specified. Several accepted standard generating polynomials exist. A standard 32 bit generating polynomial is defined for the Autodin II and Ethernet^{tm} standard. This generating polynomial is found in the draft proposed American National Standard for FDDI Media Access Control X3T9.5/83-16 update of 6-01-84. This standard generator polynomial is:

$$(2) \quad P(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

Cyclic check or CRC bit computation involves dividing the message polynomial by the generator polynomial to generate a quotient polynomial and a remainder polynomial. The quotient polynomial is discarded and the remainder polynomial coefficients are appended to the message polynomial as CRC check bits.

The combined message and check bits are then transmitted over the communication link and arrive at the receiver either modified or unmodified depending upon whether errors occurred during the transmission. Generally, the receiving apparatus divides the complete received message, including check bits, by the same generator polynomial which was used to generate the check bits at the transmitter end of the link. The result of this division is a zero remainder polynomial if no error occurred during the transmission. A non-zero remainder indicates the presence of an error.

The type of apparatus that is used to perform the above described calculation on serial format input data is shown in Figure 1. Figure 1 is a block diagram of a CRC checkbit calculation machine. The CRC checksum register 30 is comprised of a plurality of memory cells that store the CRC check bits. The outputs of these memory cells are coupled to the inputs of an array 32 of shifting links some of which are exclusive-OR gates as best seen in Figure 2.

Figure 2 is a detailed circuit diagram of the block diagram of Figure 1. The most significant output bit from the checksum register 30 is exclusive-ORed by an input gate 34 with the incoming serial data stream of the message polynomial, and the output of the exclusive-OR gate 34 which performs this function is coupled to an input of all the other exclusive-OR gates in the array. The array shifting links which are not exclusive-OR gates are simple conductors which merely shift the incoming data one bit position left or toward the most significant bit position. The outputs of the shifting links of the array are coupled back to the data inputs of the checksum register 30 by a bus 36. The exclusive-OR gates in the array 32 as well as the straight through conductors have their outputs coupled to the input of the checksum register of the next most significant bit position relative to the bit position of the input bits positions for each shifting link from the checksum register. A bit clock signal on the line 38 clocks the raw serial format

input data of the message polynomial into the input gate 34 and causes the checksum register to load the data from the bus 36 into its memory cells. After all the raw input data bits in the message have been clocked in, the checksum register 30 contents are the CRC check bits for the message bits so processed.

Figure 3 illustrates the format of the composite data packet that is transmitted after calculation of the checkbits. The segment 40 is the message polynomial upon which the CRC bits were calculated. These message polynomial bits are transmitted simultaneously with the calculation of the CRC bits in that each time a bit is input to the gate 34, it is simultaneously transmitted. The segment 42 is the complement of the CRC bits stored in the checksum register 30 after all the bits in the segment 40 have been processed. The segment 42 is comprised of complement CRC bits so that when CRC bits are calculated on the receiving end on the combined segments 40 and 42, the remainder will come out zero. In some protocols, the checksum register is preset to all logic 1's before the CRC calculation starts. In such a case, when CRC checkbits are calculated on the combined packet consisting of segments 40 and 42, the remainder will not be all zeros but will represent a standard remainder polynomial. This remainder polynomial will result every time when CRC checkbits are calculated on the combined segments 40 and 42 regardless of the bit pattern in the message polynomial 40.

The CRC checkbits in the segment 42 are sent following the segment 40 by switching a multiplexer 44 with a select signal on the line 47 to deselect the serial data input line 46 and select the output line 48 of an inverter 50. The input of the inverter 50 is coupled to the output of the most significant bit position memory cell in the checksum register. The inverter 50 inverts the check bits as they are clocked out in serial fashion by the bit clock signal on the line 38. The composite packet comprised of segment 40 followed by the CRC checkbits 42 appears on the serial output line 52.

A problem arises with the architecture of Figure 1 where no bit clock signal is available to clock in raw input data to the input gate 34. Some systems are byte oriented and only provide a byte clock signal for every eight bits. Such systems must be able to compute CRC bits by accepting one byte of raw input data at a time and simultaneously computing the CRC bits taking into account the effect of each bit in the raw input data byte. An architecture to accomplish this parallel CRC computation is shown in Figures 4A and 4B.

In Figures 4A and 4B the array of shifting links is comprised of a plurality of rows of shifting links with each row assigned to process one of the bits of the raw input data byte. The raw input data byte is shown as the bits D7 through D0 coupled to the input gates on the left. Each of these input data bits is coupled to an input of one of the input exclusive-OR gates 67, 69, 70, 72, 74, 76, 78 and 80. Each of these input gates has its output coupled to an input of each exclusive-OR gate in its row and to the input of the least significant bit position shifting link in the next row. Thus each row in the array 56 acts like the shifting links row 32 in Figure 1 except that its outputs are connected to the inputs of the next row. The first row has its inputs coupled to the outputs of the checksum register 30 and the last row has its outputs coupled to the inputs of the checksum register. Each row has one input of its input gate coupled to the output of one of the bits in the highest order byte in the checksum register, the first row being connected to the highest order bit and the second row being connected to the second most significant bit and so on for all the rows. Each shifting link in each row shifts its input bit one bit position toward the most significant bit position of the checksum register. The architecture of Figure 4 therefore calculates CRC bits by processing 8 bits of raw input data at a time.

If the architecture of Figure 4 is to be integrated, several improvements can be made which save chip area and which enable the architecture to perform several functions which cannot be done with the architecture of Figure 1. For example, to get the CRC bits out of the checksum register 30 in parallel format in the architecture of Figure 1 or Figure 4 requires that a conductor be connected to each output of the checksum register 30. For a 32 bit checksum register, this would require that a great deal of chip area be consumed by the output bus conductors. It would be useful if only the highest order byte of CRC outputs were connected to the output bus and the other bytes of CRC data were shifted into the highest order byte for output. This would cut down the number of conductors in the output bus from 32 to 8 thereby saving much chip area.

In some systems it is necessary to calculate a first set of CRC bits on a first data packet and a second set of CRC bits on a second data packet either immediately following the first data packet or immediatly following transmission of the CRC checkbits calculated on the first data packet. It is customary in many CRC calculation machines to preset the contents of the checksum register to all logic 1's just before the start of the CRC calculation. This improves the performance of the CRC calculation machine in that input data strings having long strings of logic 0's will still affect the contents of the checksum register such that if there is a malfunction in the checksum register or the

calculation array, the malfunction can be immediately detected. If the checksum register were not preset to all 1's, a defect in the checksum register or calculation array might not be detected in such a circumstance.

When separate CRC check bits are to be calculated on two back to back packets, there is no clock cycle between the first and second packets during which the checksum register can be preset by inputting all logic 1's into the memory cells. It would be useful to provide a way to preset the CRC calculation machine for the second data packet in such a situation where there is no spare clock cycle between the first packet and the second packet.

It is common in networks of computers to formulate data packets which have header bits which define the network and the particular node of that network to which is addressed a data message appended to the header bits. It is desirable in such situations to be able to calculate the CRC bits in either of two ways. The first way is to calculate a header CRC on the header bits, and then to calculate a data CRC on the data message. The second way is to calculate a header CRC on the header bits, and then to calculate a data CRC on the whole packet including the header bits, the header CRC check bits and the data message. It would be useful to provide a CRC calculation machine which could calculate CRC checkbits using either of these two methods.

In the token ring computer network environment where all nodes in the network are connected together in a ring by a single cable, it is common to have multibyte messages being sent along the network where the first byte has certain initial bits which can be changed on the fly by any node in the network. It not desirable to include these initial bits which are subject to unpredictable changes which are not errors in the CRC calculation. Such changes would show up in the CRC check bits as errors if a change in one of these bits occurred while passing through a node between the transmitting and the receiving node. It would be useful to provide a CRC calculation machine which could calculate the CRC on a data packet using only a byte clock and still be able to exclude from the CRC calculation any number of initial bits in the first byte of a message.

According to the present invention there is provided a CRC calculation apparatus comprising means for storing a plurality of CRC bytes, and calculation means for accepting input data from the storage means and raw input data, the calculation means dividing the raw input data by a predetermined binary number and storing the remainder of the division in the storage means as CRC check bits, characterised by shifting means coupled to the storage means and the calculation means for shifting the CRC bytes one byte at a time in the storage means toward a predetermined byte position in synchronization with a byte clock signal by deactivating the calculation means and sending the current contents of the storage means through the calculation means without change while simultaneously shifting the CRC bytes and then storing the unchanged, shifted CRC bits back in the storage means at positions which are shifted one byte toward the predetermined byte position, the foregoing actions being performed once in response to each activation of the byte clock signal, and a CRC output bus having one conductor coupled to each respective data output of the storage means for each CRC bit in the CRC data byte stored in the predetermined byte position, whereby the CRC bytes are made accessible for output via said output bus one byte at a time.

In the present specification there is described by way of example a CRC calculation machine for 32 bit CRC checkbit packets which can output the CRC data in parallel format without the need for a 32 bit wide output bus. The machine uses an 8 bit wide output bus which has one conductor coupled to the data output of each memory cell storing a CRC bit in the most significant byte in the checksum register. The clock inputs of the checksum register memory cells are coupled to a byte clock signal which causes loading of data at the inputs of the checksum register periodically. The inputs of the checksum register are coupled to the outputs of the last row of an array of shifting links comprised of one row of shifting links for each bit of a raw input data byte. Each row has one column or shifting link position for each bit in the checksum register. Some of the shifting links are exclusive-OR gates which have one input coupled to the output of an input gate. This input gate performs an exclusive-OR function between one bit of the raw input data byte and one bit of the most significant byte of data stored in the checksum register. The other input of each exclusive-OR gate is coupled to an output bit of the checksum register for the column in which that particular gates resides. The outputs of the exclusive-OR gate shifting links are coupled to the input of the shifting link in the next row in the next most significant bit position. The shifting links which are not exclusive-OR gates are conductors which conduct the bit at their inputs to the input of the shifting link in the next row in the next most significant bit position, i.e., they perform a simple one bit shift function.

The shift function of each shifting link is as follows. If the most significant bit in th checksum register is stored in the leftmost cell, each shifting link shifts its bit one bit to the left. This shifting function is used to advantage in the invention to

implement the one byte wide output bus while maintaining the capability to output each byte in the checksum register in parallel format.

After the CRC calculation is completed, the most significant byte of CRC data is read by simply reading the data on the output bus. The remaining bytes are read out one byte at a time in synchronization with the byte clock signals by using the array of shifting links to perform simple 8 bit left shifts for each byte clock signal. This is done by adding an AND gate to the shifting link array between the output of each input gate and the inputs of each exclusive-OR gate shifting link in each row. These AND gates have one input coupled to the output of the input gate and have another input coupled to a SHIFT not signal. The AND gates have their outputs coupled to all exclusive-OR gate inputs in the array row to which the output of the input gate was formerly connected. The SHIFT not signal, when asserted low, causes each exclusive-OR gate in the row in the array of shifting links to become transparent. This results in the data in the checksum register being passed through each row and shifted left by one bit position. After passing through each row in the array, the data will have been shifted 8 bits to the left for each byte clock signal. Thus upon the occurrence of the first byte clock signal when the SHIFT not signal is asserted, the second most significant CRC byte will be shifted into the memory cells of the most significant byte and may be read from the output bus. Upon the occurrence of the next byte clock signal, the third most significant byte of CRC data will be shifted from the former position of the second most significant byte into the position of the most significant byte. This process continues until all the bytes of CRC data have been shifted into the position of the most significant byte and read from the output bus.

In the accompanying drawings, by way of example only:-

Figure 1 is a block diagram of a serial format CRC calculation machine using a bit clock.

Figure 2 is a logic diagram for the machine of Figure 1.

Figure 3 is a diagram of the data format for a typical data message with its CRC bits appended.

Figure 4 (4A + 4B) is a logic diagram for a parallel format CRC machine using a byte clock.

Figure 5 is a block diagram of a CRC calculation machine embodying the invention and using a single byte wide output bus for outputting the CRC bytes.

Figure 6 is a block diagram of a CRC calculation machine embodying the invention and using a single byte wide output bus connected to each byte of CRC data through a multiplexer.

Figure 7 (7A +7B) is a logic diagram of the logic of the embodiment of Figure 5 showing the AND gate used in the array of shifting links to cause the shifting of lower order CRC bytes into the most significant byte position for output.

Figure 8 is a block diagram of a circuit for presetting the checksum register with all 1's using an input multiplexer and presetting in the clock cycle before the first clock cycle of the CRC calculation.

Figure 9 is a block diagram of a serial input data format circuit for presetting the checksum register with all 1's using OR gates and presetting in the first clock cycle of the CRC calculation.

Figure 10 is a block diagram of a parallel input data format circuit for presetting the checksum register with all 1's using OR gates and presetting in the first clock cycle of the CRC calculation.

Figure 11 is a block diagram of a serial input data format circuit for presetting the checksum register with all 1's using multiplexers and presetting in the first clock cycle of the CRC calculation.

Figure 12 is a block diagram of a parallel input data format circuit for presetting the checksum register with all 1's using multiplexers and presetting in the first clock cycle of the CRC calculation.

Figure 13 is a timing diagram of the control signals which control the preset function in the circuits of Figures 8 through 12 in either the serial or parallel format input data format and in either the circuits where the presetting is done in the clock cycle preceding the first clock cycle of the CRC calculation or during the first clock cycle of the CRC calculation.

Figures 14A and 14B illustrate the data packet format of a typical network message and illustrates two different methods of calculating CRCH checkbits on the header and CRCD checkbits on the entire message or on the data packet alone.

Figure 15 illustrates one serial format input data circuit for calculating the CRCH and CRCD checkbits according to either of the methods illustrated in Figure 14.

Figure 16 illustrates one parallel format input data circuit for calculating the CRCH and CRCD checkbits according to either of the methods illustrated in Figure 14.

Figure 17 illustrates a preferred serial format input data circuit for calculating the CRCH and CRCD checkbits according to either of the methods illustrated in Figure 14.

Figure 18 illustrates a pr ferr d parallel format input data circuit for calculating the CRCH and CRCD checkbits according to either of the meth-

ods illustrated in Figure 14.

Figure 19 illustrates the timing for the control signals which control the embodiments of Figures 17 and 18 in calculating the CRCH and CRCD checkbits according to methods illustrated in Figure 14.

Figure 20 (20A + 20B) illustrates the variable bit boundary and several other features of a CRC calculation machine.

Figure 21 is a timing diagram for the embodiment of Figure 20 illustrating the operation of the variable bit boundary function.

Regarding the drawings, it should be noted that the embodiments shown in Figs. 8-12 are the subject of our copending application 86309175.7 (EP-A-0225763); the embodiments shown in Figs. 15-19 are the subject of our copending application 86309177.3 (EP-A-0230730); and the embodiment shown in Figs. 20-21 is the subject of our copending application 86309147.6 (EP-A-0225761).

BYTE WIDE OUTPUT BUS

Turning to Figures 5 and 7A and 7B there are shown, respectively, block diagrams and logic diagrams of the preferred embodiment of a CRC machine for calculating multiple CRC bytes and outputting them over a single byte wide output bus. The embodiment of Figure 5 employes a checksum register 30 comprised of a plurality of memory cells, each having a data input, a data output and a clock input for receiving a byte clock signal. Upon receipt of the byte clock signal, any data on the inputs of the memory cells will be latched into the cell and reflected on the data outputs of the cells. Each memory cell operates independently of the adjacent cells on its left and right. The checksum register is segmented logically into a plurality of bytes of CRC data designated bytes 0 through 3 in Figure 5. In the preferred embodiment, there are 8 bits per byte and 32 bits total in the checksum register with byte 3 being the most significant byte. In Figure 5, byte 3 is comprised of memory cells 24 through 31.

The data outputs of the checksum register are coupled to the inputs of an array of shifting links 60 as shown in Figure 7A and 7B. This array of shifting links is comprised of one row of shifting links for each bit of an raw input data byte comprised of the data bits D0 through D7 which enter the array on the left. These raw input data bits are the data bits of the message upon which a CRC calculation is desired. Each raw input data bit is coupled to one input of an input gate. There is one input gate designated for each row with a designated raw input data bit designated for that row and that input gate. For the first row, the designated raw input data bit is D7 and the designated input gate is the exclusive-OR gate 62. For the second row, the designated raw input data bit is D6, and the designated input gate is the exclusive-OR gate 64. A similar situation exists for each row and for all the bits of the raw input data byte. If the data bits were arriving in serial format, most significant bit first (an arbitrary assumption) and D7 was designated the most significant bit (another arbitrary assumption), then the designated raw input data bit for the first row would be the most significant bit of each bit or the first arriving bit if the bits were arriving serially. The designated bit for the second row would be the second most significant bit or the second bit to arrive if the bits were arriving in serial format. The third row would have as its designated bit, the third most significant bit or the bit arriving third in time and so on for all the rows.

Each input gate has another input coupled to one of the data outputs of one of the bits of the most significant CRC byte. The input gate for the first row, gate 62 has its other input coupled to the data output of the most significant CRC bit, bit 31. The input gate for the second row has its other input coupled to the second most significant CRC bit, bit 30 and so on for all the rows.

Each row of shifting links is comprised of a plurality of straight through conductors which do nothing but shift the bits at their inputs to their outputs which are connected to the inputs of the next row of shifting links in the next most significant bit position. That is these straight through conductors do nothing but shift their input bits one bit position left. The balance of the shifting links in each row are exclusive-OR gates which have one input as a bit input for the CRC bits from the checksum register or from the preceding row and which have another input coupled to the output of the input gate designated for that row. This latter input is not coupled directly to the output of the input gate, but is coupled to the output of an AND gate designated for the particular row. One input of this AND gate is coupled to the output of the exclusive-OR gate and another input is for receipt of a SHIFT not signal the purpose of which will be explained below. The output of the AND gate for each row is also coupled to the input of the least significant shifting link of the next row. There is one designated AND gate for each row, e.g., gate 66 for the first row and gate 68 for the second row.

The output of each exclusive-OR gate shifting link is coupled to the input of the next most significant bit position shifting link in the next row, i.e., each exclusive-OR gate shifts its input bit one bit position to the left after performing the exclusive-OR operation on it.

The relative positions of the exclusive-OR gate shifting links depends upon the particular generator

polynomial being used. For the Autodin II and Ethernet standard generator polynomial given in formula (2) above, the exclusive-OR gates must be located so as to have their inputs coupled to bits 0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22, 25 and 31 of the checksum register 30. The reason for this is well known to those skilled in the art of CRC calculation and will not be explained here for brevity and simplicity. Each row has its exclusive-OR gates in the same bit position. The inputs of the first row are coupled to the outputs of the checksum register as if it were a preceding row, and the outputs of the last row are coupled to the inputs of the checksum register as if it were the next row.

The data outputs of the most significant byte of the CRC register are also coupled to the individual conductors of the CRC output bus 70. Preferably, each memory cell will have a complement output, and it is these outputs of the most significant byte which are coupled to the output bus 70. Alternatively, an inverter can be used in each line to invert the CRC bits prior to transmission.

The purpose of the AND gates and the SHIFT not signal is to allow the array of shifting links to be made transparent so that data at the output of the checksum register will pass through the array unchanged and be shifted left by one byte in the process and re-input into the checksum register. When the SHIFT not signal is asserted, i.e., logic 0, all the AND gates such as the gates 66 and 68 have logic 0 outputs. The logic 0's on the lines 63 and 73 through 79 renders the exclusive-OR gates in the rows of shifting links transparent such that they pass their input data from the preceding row through to their outputs coupled to the next row without change. Because each row shifts its input data one bit to the left, and because there are 8 rows, the result is that for every cycle of the byte clock signal the data in the checksum register is shifted left by one byte when the SHIFT not signal is asserted. This allows all bytes of the CRC data to be accessed through the single byte wide output bus 70.

The embodiment of Figure 5 may also be used in a serial format with some modifications. If the array of shifting links 84 is a single row of shifting links, then the AND gates of Figure 7 will not work to cause single byte left shifts for every byte clock signal. In a serial data input environment, the raw data input to the array 84 would be one bit at a time in serial format in synchronization with a bit clock signal. After all the raw input data bits are processed, the CRC checkbits for the message would reside in the checksum register 30. The most significant byte could be read immediatly on the output bus 70. The remaining bytes would be shifted left one bit at time by an additional coupling of each CRC bit data output to the input of the next most significant CRC bit input, i.e., the input of the neighboring cell on the left. This coupling would be through a multiplexer for each bit which would have one input coupled to the bus 36 and the other input coupled to the CRC bit data output on the right. The output of the multiplexer would be connected to the data input of the left neighbor. When outputting CRC bytes 2, 1 and 0, these multiplexers would be set to select the inputs coupled to the right neighbor of each cell. An additional logic circuit would then disable the output bus 70 for 8 cycles of the bit clock signal connected to the clock input of each cell in the checksum register and enable it when byte 2 had been shifted into the byte 3 or most significant byte position. The same process would be performed for each byte until all bytes had been read out.

Referring to Figure 6 there is shown another embodiment for an output bus structure for a CRC calculation machine. This embodiment uses a multiplexer having 4 inputs each of which is coupled by a byte wide bus to the complement data outputs of one group of memory cells storing one byte of CRC data. A CRC not byte select signal on a bus 74 selects which of the four inputs to couple to the 8 bit output bus 70. CRC calculation is performed as described above, and it does not matter with respect to outputting the CRC data whether the array of shifting links 84 is a serial array or a parallel array or whether a bit clock or byte clock signal is used to clock the data in except the type of array and clock signal must be consistent.

PRESET EMBODIMENTS

As mentioned above, it is accepted practice to preset the checksum register to all logic 1's before the start of the calculation of CRC checkbits. Figures 8 through 12 illustrate various circuits for performing the preset function in two different ways and in both the serial and parallel calculation environment. Figure 8 shows the circuit of a CRC calculation machine using preset apparatus which must be actuated at least one clock cycle before the first clock cycle of the CRC calculation. The CRC apparatus is comprised of a checksum register 30 having its data outputs coupled to the data inputs of an exclusive-OR and shift array 84 as in the embodiments described above. The array 84 can be serial as the array 32 of Figures 1 and 2 or it can be parallel as is the array 60 of Figures 4A and 4B or 7A and 7B. In fact any of these arrays will suffice for purposes of practicing this embodiment of the invention as will any other array of shifting links described hereafter. As in the above described embodiments, the last row in the array is coupled through a multiplexer 86 to the data inputs of the checksum register 30 by a data bus 36

which is 32 bits wide.

The multiplexer 86 has three 32 bit wide inputs marked A, B and C, and a 32 bit wide output bus 88 which is coupled to the data inputs of said checksum register. The multiplexer has three select inputs for receiving input select signals. The particular select signal which is active at any particular time causes its corresponding one of inputs A through C to be coupled to the output bus 88. The A input is 32 logic 1's, and the B input is the 32 data outputs of the checksum register collectively referred to as bus 91. The C input is the 32 bit bus 36 carrying the data outputs from the last row in the array 84.

The input select signal PRESET SELECT is activated one or more clock cycles before the first clock cycle of the CRC calculation. This causes 32 logic 1's from input bus A to be coupled to the the bus 88 and loaded into the cells of the checksum register 30. Next, the COMPUTE SELECT input is asserted to couple the C input or bus 36 to the bus 88. While this condition exists, CRC calculation will proceed as the data of the message for which check bits are to be generated is clocked into the array 84. If at any time it is desired to stop the CRC calculation, the HOLD SELECT input may be asserted thereby connecting the outputs of the checksum register back into the inputs via the buses 90 and 88.

Figures 9 through 12 illustrate various embodiments of a CRC calculation machine where the preset apparatus allows the preset logic 1's to be loaded during the first clock cycle of the CRC calculation. Each of these embodiments, and all the other embodiments described herein which do not use a feedback bus such as the bus 130 in Figure 17, employs an AND gate 100 in the clock line. These AND gates perform a logical AND operation between the clock signal and a COMPUTE signal which is asserted as a logic 1 when it is desired to compute CRC checkbits. When COMPUTE is a logic 1, CRC calculation can proceed. When COMPUTE is a logic zero, no CRC computation occurs, and the contents of the checksum register are constant.

Figures 9 and 10 show serial array and parallel array embodiments respectively using OR gates to implement the preset function. Figures 11 and 12 show serial and parallel array embodiments respectively using multiplexers to implement the preset function. Of these embodiments, Figure 10 is the preferred embodiment. All of these embodiments will be discussed simultaneously as to the preset apparatus since the elements and the operation of the various elements and the form of the CRC calculation is the same for all the embodiments. These elements operate in the same manner as in the serial and parallel array embodiments of other embodiments of the invention discussed above. Further, the arrays themselves can be any of the serial or parallel arrays described herein for purposes of practicing the embodiments of the invention regarding preset during the first clock cycle of the CRC calculation without adversely affecting operation of this embodiment of the invention.

An important element in each of the embodiments of Figures 9 through 12 is the coupling of the data outputs of the checksum register to the data inputs of the array of shifting links through circuitry which forces all the data inputs of the array of shifting links to a logic one state during the same clock cycle that the CRC calculation starts. The embodiments of Figures 9 and 10 use OR gates to implement this function of forcing logic 1's at the desired time while the embodiments of Figures 11 and 12 use multiplexers. In Figures 9 and 10 each of the four OR gates 87,89,92 and 93 represents 8 OR gates, and each performs an OR logic function on the bits of one of the bytes of CRC data stored in the checksum register 30. Each of the 8 OR gates represented by each of the gates 87,89,92,93 has an input coupled to one of the data outputs of a memory cell in the particular group of cells in the checksum register coupled to the corresponding OR gate 87,89,92,93. The output of each OR gate is coupled to an input of one of the shifting links in the first row of the array. Each OR gate has another input coupled to a NEW PRESET signal. When this signal is asserted as a logic 1, all the outputs of the OR gates assume a logic 1 value thereby forcing logic 1's at the inputs of the array of shifting links during the first clock cycle of the CRC calculation. This situation is equivalent to the situation which would exist if the checksum register 30 had been loaded with logic 1's on a previous clock cycle and those logic 1's were transmitted to the inputs of the first row of the array of shifting links on the first clock cycle of the CRC calculation.

After the first clock cycle of the CRC calculation, the NEW PRESET signal reverts back to the logic 0 state thereby rendering the OR gates transparent. Thereafter any data in the checksum register is transmitted through the OR gates 90 through 93 without change, and the CRC calculation proceeds normally.

Figures 11 and 12 accomplish the same forcing of logic 1's into the inputs of the first row of the array of shifting links using multiplexers 94 through 97 to redirect the inputs of the first row of shifting links in the array 32 or 60 to a source of 32 logic 1's. The multiplexers 94 through 97 could be represented as one multiplexer having two 32 bit inputs. One input would be connected to each of the 32 data outputs of the cells in the checksum register

30, and the other input would be for coupling to a source of 32 logic 1's such as a voltage source coupled to each of the conductors of the input. The output of the multiplexer would be a 32 bit bus coupled to the data inputs of the first row of shifting links in the array. The multiplexer would have an input for receiving a NEW PRESET SELECT signal. This signal is asserted during the first clock cycle of the CRC calculation and causes the multiplexer to select the input coupled to the source of logic 1's and to couple these logic 1's to the inputs of the first row of shifting links in the array. After the first clock cycle of the CRC calculation, the NEW PRESET SELECT signal reverts back to the state wherein the multiplexer couples the data outputs of the checksum register 30 to the data inputs of the array. Thereafter CRC calculation proceeds normally.

Figure 13 shows a timing diagram showing the relationships of the clock signal and all the control signals of the embodiments of Figures 8 through 12. This diagram is self explanatory given the above discussion of the relationships of the present signals to the first clock cycle of the CRC calculation.

HEADER CRC CALCULATION

In many applications for transmission of serial data, header packets are used. These header packets are bits which define the addressed node for which the data packet attached to the header packet is intended. Such message organization is commonly used in networking of computers, terminals and peripherals. Figures 14A is a symbolic drawing of a typical message organization where CRCH 101 is a collection of CRC checkbits calculated on the header bits 100 and where CRCD 103 is a collection of CRC checkbits calculated on the data message 102.

There are commonly used two different methods of calculating the CRCD checkbits. The first way is shown in Figure 14A where the CRCD checkbits are calculated on the data message 102 alone. The second way is shown in Figure 14B where the CRCD checkbits are calculated on the entire packet including the header 100, the CRCH checkbits 101 and the data bits 102. It is useful to have a CRC calculation machine which is capable of calculating separate CRCH and CRCD checkbit packets in either of the two ways.

Any of the embodiments described above or to be described below which have a gate in the feedback bus 36 or which have the hold bus such as the bus 91 from the output of the checksum r gister 30 to its input through an input multiplexer 86 can calculate separate CRCH and CRCD checkbit packets according to the method of Figure 14A. This is done by stopping the CRC calculation after the header packet has been processed, outputting the then existing contents of the checksum register 30 as the CRCH packet and then presetting the checksum register and allowing the CRC calculation to begin again when the data message 102 begins to arrive.

The embodiment of Figure 15 is one embodiment of a CRC calculation machine which can compute separate CRCH and CRCD checkbit packets in accordance with the method of Figure 14B using a serial format array 32 of shifting links. The embodiment of Figure 16 is one embodiment of a CRC calculation machine which can compute separate CRCH and CRCD checkbit packets in accordance with the method of Figure 14B using a parallel format array 32 of shifting links. Both of these embodiments use a separate snapshot register 106 and a multiplexer 108 to implement the method of Figure 14B. The common portions of these embodiments will be described below without distinguishing between whether a serial or parallel array of shifting links is used since this is irrelevant to the operation of this aspect of the invention.

A checksum register 30 identical in construction, operation and purpose to the checksum registers described above in other embodiments has its data inputs coupled to the data outputs of an input multiplexer 86 by a 32 bit wide bus 88. The multiplexer 86 has three inputs which are the same inputs and which serve the same purpose of the inputs for the multiplexer 86 in Figure 8 discussed above.

The output of the checksum register 30 is coupled to the input of the array of shifting links either directly as in Figure 15 or through a preset means 109 as shown in Figure 16. The embodiment of Figure 15 uses the preset method where the logic 1's are loaded into the checksum register 30 during the clock cycle before the CRC calculation starts. This is done by asserting the INPUT SELECT signal so as to select the input C of the multiplexer 86 during the clock cycle before the start of the CRC calculation so as to load 32 logic 1's into the checksum register 30. The embodiment of Figure 16 uses the preset method of forcing logic 1's into the input of the array of shifting links during the first clock cycle of the CRC calculation method as described above with respect to Figures 9 through 12. The preset means 109 can be either OR gates like the OR gates 87,89,92,93 in Figures 9 and 10, or multiplexers like the multiplexers 94 through 97 shown in Figures 11 and 12. Either preset method and apparatus may be used on either of the embodiments of Figures 15 or 16.

The array of shifting links 32 or 60 can be any of the arrays of shifting links described herein. Regardless of whether the array 32 or 60 is for

processing of serial format data messages or parallel format data messages, the outputs of the array are coupled to 32 data inputs of a snapshot register 106 which serves as a storage location for a copy of the CRCH checkbits. The snapshot register is comprised of a plurality of memory cells like those used in the checksum register, but those skilled in the art will appreciate other types of memory cells which may be used. The clock inputs of the memory cells in the snapshot register 106 are for coupling to an END OF HEADER clock signal which serves to load the output of the array of shifting links 32 or 60 into the snapshot register 106 after all the header bits have been processed and the output of the array on the bus 36 consists of the CRCH bits. These CRCH bits can be output on the CRCH output bus 110 byte by byte by use of a multiplexer such as the multiplexer 72 in Figure 6 or in other methods which will be apparent to those skilled in the art.

To implement the CRCD calculation method of Figure 14B, the CRCH bits must be fed back into the array of shifting links as raw input data so that CRC bits may calculated on the CRCH bits. This is the purpose of the multiplexer 108 and the circuitry connected to it. The structure of this circuitry will be described briefly here followed by a description of its operation in implementing the method of Figure 14A and in implementing the method of Figure 14B.

In the embodiment of Figure 15, the snapshot register 106 is connected as a shift register so that the CRCH bits may be shifted into one input of the multiplexer 108 in serial fashion on a line 112. The other input of the multiplexer 108 is coupled to the source of the serial raw input data. A select signal on a line 114 causes the multiplexer 108 to couple either the raw data in the message on line 116 or the CRCH data on the line 112 into the raw data input 118 of the array 32.

A similar situation exists for the embodiment of Figure 16 except that the snapshot register 106 has 4 one byte wide output buses 118 through 121 which are coupled to 4 input ports of a multiplexer 116. The parallel format input data bits D7 through D0 are coupled by a bus 122 to an input port of the multiplexer 116, and a BYTE SELECT signal on a line 124 controls which of the inputs of the multiplexer 116 are to be coupled to the raw data input 126 of the array of shifting links.

The embodiments of Figures 15 and 16 can be used to calculate the CRCD checkbits in the manner of Figure 14A. First, a preset operation must be done. Specifically as to the Figure 15 embodiment, by asserting the INPUT SELECT signal to select input C, 32 logic 1's will be loaded into the checksum register 30 in preparation for the CRC calculation. The Figure 16 embodiment can perform the preset in the same manner if the input multiplexer 86 is modified in Figure 16 to have the inputs of Figure 15, and the preset operation will be performed in the same manner. Either of the Figure 15 or Figure 16 embodiments can be preset using the new preset means 109 by asserting the NEW PRESET signal on the line 126 in the manner previously described. Next, by asserting the INPUT SELECT signal to select the A input of the multiplexer 86, CRC calculation can begin. The input A will be selected during CRC calculation on the header bits 100.

After processing of the header bits, if the method of Figure 14A is to be used to calculate the CRCD checkbits, the input B will be selected during the clock cycles when the CRCH bits are output from the checksum register 30. This output process for outputting CRCH can be by any of the methods described above for serial or parallel shifting arrays depending upon the type of output bus structure coupled to the snapshot register.

After the CRCH bits are output, the checksum register is preset to all 1's again if the method of Figure 14A is to be performed. The raw input data of the data message 102 is then input to the array 32 or 60 and processed to calculate the CRCD checkbits.

If the method of Figure 14B is to be performed, the presetting steps and calculation steps are the same as above for calculating CRCH. However, upon completion of calculation of the CRCH checkbits, they must be copied into the snapshot register 106 so that they may be output while simultaneously being fed back into the raw data input of the array of shifting links. To do this, the END OF HEADER signal is asserted, which causes the CRCH bits to be loaded into the snapshot register 106. The END OF HEADER clock signal can be the clock signal supplied to the checksum register, i.e., the BIT CLOCK signal, gated through a gate which only allow it to pass when the end of the header is detected. This causes a copy of the CRCH bits to be made in the snapshot register 106 and shifted out one bit at a time on the bus 112. Another copy of the CRCH bits will be input at the end of the header into the checksum register 30 via the bus 36. In the case of the embodiment of Figure 15, the multiplexer 116 is caused by the SELECT signal on the line 114 to couple the serial data stream on the line 112 to the raw data input 118. Thus the CRCH checkbits are input to the array and operated upon the fashion described previously. After all the CRCH checkbits have been processed, the multiplexer 108 is caused by the SELECT signal to switch its output 118 back to the data message bits 102 on the line 117. After all the data bits in the message 102 have been processed, the CRCD checkbits will reside in the checksum register 30

and can be output in any of the manners described above.

The preferred circuit for performing the CRCH and CRCD checkbit calculations is shown in Figure 17 in serial array format and in Figure 18 in parallel array format. Each circuit uses an input multiplexer 86 having its data outputs coupled to the 32 data inputs of the checksum register 30 by a bus 88. Each multiplexer 36 has an A input coupled to the feedback bus 36 coupled to the output of the array of shifting links 32 or 60. Each multiplexer also has an input C coupled by a bus 130 to the data outputs of the checksum register 30. Finally, each multiplexer 86 has a B input coupled to a remainder polynomial bit pattern. This bit pattern represents the coefficients of the standard remainder polynomial which results when CRC checkbits are calculated on a data message and its checkbits starting from a preset state of all logic 1's.

The arrays of shifting links can be any of the array structures described herein. The output of the array of shifting links is coupled to the input of the checksum register 30 through the input multiplexer 86 by the feedback bus 36. The input of the arrays 32 or 60 are coupled to the data outputs of the checksum register 30 through a preset means 109 which has the same structure, operation and purpose as the preset means previously described. The data outputs of the checksum register 30 are also coupled to an output bus 132 which can have any of the structures and and be used in any of the manners described above to output CRC data from the checksum register.

The manner in which the circuits of Figures 17 and 18 work to calculate the CRCH checkbits and to calculate the CRCD checkbits is best understood by referring to the timing diagram of Figure 19.

FIGURE 14A METHOD

To calculate CRCH and CRCD in the manner of Figure 14A, the arrays of shifting links in the embodiments of Figures 17 and 18 are preset to all logic 1's by asserting the NEW PRESET signal during the same clock cycle that the first bit or byte of the header packet arrives as shown on time line 2 of Figure 19. In other embodiments which use the alternative types of preset apparatus, the input select signal OLD PRESET is asserted to select the input of the input multiplexer which is coupled to a source of 32 logic one's.

Next, the COMPUTE input select signal is asserted causing the input A to be selected and the CRCH checkbits are calculated on the header bits. After all bits of the header packet 100 have been processed, the CRCH checkbits will reside in the checksum register and may be shifted out on the CRC output bus in the case of the embodiment of Figure 17 or may be output one byte at a time using the apparatus described earlier herein. This process is symbolized by the assertion of the signal MUX or SHIFT on time line 4 of Figure 19. In alternative embodiments, all 32 bits of CRCH checkbits may be output in parallel. During the time when the CRCH bits are being output, the input multiplexer 86 is caused by the assertion of the HOLD signal illustrated on time line 6 to select input C for coupling to the checksum register input. This maintains the contents of the checksum register constant during the time the CRCH bits are being output to implement the method of Figure 14A.

Upon completion of the output of the CRCH bits, the checksum register is again preset with all 1's by either asserting the NEW PRESET signal as shown on time line 7 of Figure 19 or by asserting the OLD PRESET signal during the clock cycle before the data message starts as shown on time line 8 of Figure 19. Thereafter, the input multiplexer is caused by assertion of the COMPUTE signal to again select the A input for coupling to the input of the checksum register 30. This causes the CRC calculation to resume on the raw input data bits of the data message 102 arriving at the raw data inputs 123 and 126. After all the data bits in the message 102 have been processed, the CRCD checkbits will be present in the checksum register 30. That concludes the method of calculation of the CRCH and CRCD checkbits per the method of Figure 14A.

FIGURE 14B METHOD

To calculate the CRCH and CRCD checkbits in accordance with the method of Figure 14B, the embodiments of Figures 17 and 18 are operated as follows. The NEW PRESET signal is asserted during the first clock cycle of the header as before and as shown on time line 2 of Figure 19. The COMPUTE input select signal is simultaneously asserted as shown on time line 1 of Figure 19. This selects the input bus A of the input multiplexer for coupling to the input of the CRC checksum register. The CRCH calculation then proceeds as earlier described. The CRCH checkbits will be present in the checksum register after all the header bits have been processed.

As before, these CRCH checkbits must be held constant in the checksum register 30 while they are output. However, there is no snapshot register to hold a copy of them, so the input multiplexer 86 must be forced by assertion of the HOLD signal to select the input C so the CRCH bits in the checksum register 30 are recirculated without change until they have all been output. The outputting process for the CRCH bytes is identical to the

process as described above and is symbolized by the signal assertion shown on the time line 4 of Figure 19.

Several clock cycles will have passed during this process of outputting the CRCH bits without any CRCD checkbit calculation on the CRCH bits having been done. To implement the method of Figure 14B, the CRCD checkbits must be calculated on the entire packet including the header 100, the CRCH bits 101 and the data packet 102. To accomplish this, the input multiplexer is forced to select the input B at the end of outputting of all the CRCH bits during the clock cycle just before the first clock cycle in the input of the data portion 102 of the message 99. This is done by the assertion of the INITIATE REMAINDER signal illustrated on time line 5 of Figure 19. This loads the remainder polynomial bit pattern into the checksum register 30 at the very same time that the bit pattern would have existed in the checksum register anyway if calculation of CRCD data bits had been continuous since the first bit of the header. In other words, if the input C had not been selected at the end of the header to hold the CRCH bits in the checksum register constant during the output thereof, the bits which would have resulted in the checksum register by the time all the CRCH checkbit had been processed would be the bits in the remainder polynomial bit pattern. That this bit pattern can be accurately predicted is known since anytime a CRC calculation is performed on the data message plus the CRC bits calculated on the message alone, the resulting CRC bits represent a known polynomial which is published in the standard incorporated by reference herein defining the generator polynomial for the Autodin II and Ethernet standards.

No presetting is done before processing the bits in the data packet 102 in this method. After all the bits in the data packet 102 have been processed, the CRCD checkbits reside in the checksum register and can be output in any of the manners described herein.

VARIABLE BIT BOUNDARY CRC CALCULATION

It is common in computer networks to send multiple byte messages to other nodes on the network where the first few bits of the first byte are subject to being changed on the fly by various nodes in the system. This is especially common in token ring networks. Since these bits are subject to being changed on the fly, they must not be included in the CRC calculation lest they be falsely assumed to be errors.

One parallel format embodiment for calculating CRC checkbits on a variable number of bits in the first byte of the data message would be to make the first few rows of shifting links transparent. This would be done after presetting the checksum register to all logic 1's. The logic 1's are passed through to the first active row of shifting links coupled to the first bit of the raw input data to be included in the CRC calculation. However, the shifting action of the transparent rows of shifting links must be disabled to use this embodiment so that the least significant bits of the transparent rows are not filled with logic 0's as the logic 1's are shifted left in the transparent rows. This embodiment is more difficult to implement than the embodiment to be described next.

One preferred aspect of the invention is to provide a flexible means whereby one or more bits in the first byte of a message upon which calculation of CRC bits is desired may be ignored. The problem in a parallel format calculation using a byte clock where several initial bits are to be ignored is to preset the appropriate row of shifting links to all 1's. The appropriate row is the row which has as an input to its input gate (such as the gate 62 in Figure 7) the first data bit in the raw input data of the first byte in the message which is to be included in the CRC calculation. Figure 20 illustrates the apparatus to accomplish this function along with the apparatus implementing the other important functions of the invention.

The variable bit boundary function is implemented by the OR gates 110 through 125. The manner in which these OR gates cause the rows of shifting links coupled to the bits to be ignored to become transparent is best illustrated by example. The reader should simultaneously refer to Figure 21 which is a timing diagram of the control signals needed to implement the variable bit boundary function. The control signals needed to control the other apparatus illustrated in Figure 20A and 20B to perform the other functions of the invention which the embodiment of Figure 20A and 20B is capable of performing are as previously described herein.

To make the rows of exclusive-OR gates transparent so as to pass the preset logic 1's to the next row, a logic 0 must be applied to one of the inputs of these exclusive-OR gates in the affected rows, i.e., the rows coupled to the bits to be ignored. This is the function of the OR gates 110 through 117 and of the IGNORE signals coupled to each of these OR gates. Assume that the first three data bits D7 through D5 are to be ignored in the CRC calculation. The first clock cycle of the CRC calculation is arbitrarily assumed to be between the times t_0 and t_1 in Figure 21. During the first clock cycle of the CRC calculation starting at the transitions 131 of BYTE CLOCK signal illustrated on time line 1 of Figure 21, the CALCULATE control signal for the input multiplexer 86 is asserted to select

input A to allow the CRC calculation to begin as illustrated on time line 3. Also during the first clock cycle, the NEW PRESET signal is asserted to force all logic 1's into the data inputs of the first row of shifting links coupled to the input data bit D7 through the input gate 62 as illustrated on the time line 2. Also during the first clock cycle of the CRC calculation, but only during the first clock cycle, the IGNORE 7, IGNORE 6 and IGNORE 5 signals are asserted. Regardless of the logic state of the data bits D7, D6 and D5, logic 1's will exist on the lines 132, 134 and 136. The input gates 200, 202 and 204 thus receive two logic 1's at their inputs because the preset gates 138, 140 and 142 are forcing their outputs to logic 1's by virtue of the logic 1 state of the NEW PRESET signal. This is true regardless of the contents of the checksum register 30 at this time. The output lines of the exclusive-OR gates 200, 202 and 204 will be logic 0's therefore during the first clock cycle of the CRC calculation. The preset gates 109 are forcing all the data inputs of the first row of shifting links to which they are connected to a logic 1 state. These logic 1's are passed directly through to the second row by all the shifting links which are wires and are passed through all the shifting links which are exclusive-OR gates without inversion by virtue of the logic 0 on their common input line 63. This common input line carries the output signal from the input gate 200 after it is ANDed with the SHIFT not signal which is in a logic 1 state during the entire CRC calculation as shown on time line 4 of Figure 21.

The common input line 63 of the first row also carries the input data bit for the least significant bit position shifting link of the second row. Since this bit must also be a logic 1 for the arrangement to work properly, another OR gate 118 is interposed between the common line 63 and the data input of the least significant bit position shifting link in the second row. This OR gate 118 has one input coupled to the common input line 63 and another input coupled to the line carrying the IGNORE 7 signal. The output of this OR gate is coupled to the input of the least significant bit position shifting link of the second row of shifting links. Since the IGNORE 7 signal is a logic 1 during the first clock cycle of the CRC calculation, a one will be forced into the input of the least significant bit position shifting link of the second row.

Each row of shifting links in the array has an OR gate like the gate 118. The outputs of all these OR gates, i.e., the gates 119 through 125 is coupled to the input of the least significant bit position shifting link in the next row and each gate has one input coupled to the IGNORE signal for that row and another input coupled to the common input line for that row. Thus any number of rows can have logic 1's forced into their least significant bit position shifting links by asserting the IGNORE signal for that row. Likewise, any number of rows can be made transparent by asserting the IGNORE signal coupled to the OR gates 110 through 117 having their outputs coupled to the inputs of the input gate for the row to be made transparent.

In the particular example at hand, the IGNORE 7 through IGNORE 5 signals are asserted during the first clock cycle of the CRC calculation as shown on time lines 5 through 7 of Figure 21. This causes the first three rows of the array 60 to be transparent and to pass all logic 1's to the data inputs of the fourth row of shifting links. Thus during the first clock cycle of the CRC calculation, CRC checkbits will be calculated on all the bits of the first input byte D7 through D0 except the data bits D7 through D5. This occurs because the preset 1's are transmitted directly through the rows of shifting links coupled to the data bits D7 through D5 and so the row of shifting links coupled to data bit D4 acts like it is the first row in the array. After the first clock cycle of the CRC calculation is completed, the IGNORE 7 through IGNORE 5 signals are deactivated, and the CRC calculation proceeds normally.

Claims

1. A CRC calculation apparatus comprising means (30) for storing a plurality of CRC bytes, and calculation means (84) for accepting input data from the storage means (30) and raw input data, the calculation means (84) dividing the raw input data by a predetermined binary number and storing the remainder of the division in the storage means (30) as CRC check bits, characterised by shifting means (60) coupled to the storage means (30) and the calculation means (84) for shifting the CRC bytes one byte at a time in the storage means toward a predetermined byte position in synchronization with a byte clock signal by deactivating the calculation means and sending the current contents of the storage means through the calculation means without change while simultaneously shifting the CRC bytes and then storing the unchanged, shifted CRC bits back in the storage means at positions which are shifted one byte toward the predetermined byte position, the foregoing actions being performed once in response to each activation of the byte clock signal, and a CRC output bus (70) having one conductor coupled to each respective data output of the storage means (30) for each CRC bit in the CRC data byte stored in the predetermined byte position, whereby the CRC bytes are made accessible

for output via said output bus (70) one byte at a time.

2. An apparatus as defined in claim 1 wherein said calculation means (84) accepts input data one byte at a time in synchronization with said byte clock and simultaneously calculates CRC bits using all bits of the input data byte.

3. An apparatus as defined in claim 2 wherein said calculation means (84) includes a plurality of rows of shifting links (60), each shifting link having an input and an output, with one said row for each input data bit, and wherein each shifting link row has its inputs coupled to the outputs of the next least significant shifting links in the previous row except for the least significant shifting link in each row which has its input coupled to the output of the most significant shifting link in the preceding row and except for the first row which has its inputs coupled to said data outputs of said storage means (30) as if it were a preceding row of shifting links, and wherein the last row of shifting links has its data outputs coupled to said data inputs of said storage means (30) as if it were a preceding row of shifting links and wherein predetermined shifting links are gates which perform an exclusive-OR logical operation between their input bits and a predetermined signal.

4. An apparatus as defined in claim 3 wherein there is a predetermined signal for each said row of said shifting links and each said predetermined signal is generated by an input gate corresponding to that particular row which performs a predetermined logical operation between a predetermined one of said CRC bits and a data bit from said input data byte.

5. An apparatus as defined in claim 4 wherein the input gate corresponding to the first row is a gate (62) which performs an exclusive-OR logical operation between the most significant CRC bit stored in the storage means (30) and the first data bit which would arrive if the data were input in serial fashion, and wherein the input gate corresponding to the second row is a gate (64) which performs an exclusive-OR operation between the next most significant CRC bit stored in the storage means (30) and the second bit which would arrive if said input data were input in a serial fashion, and so on for each row of said shifting links.

6. The apparatus of claim 5 wherein said shifting means is a plurality of AND gates (66, 68), one for each row of said shifting links, each AND gate having one input coupled to the output of the exclusive-OR gate for a particular row and having its output coupled to one of the inputs of the exclusive-OR gates in the particular corresponding row of shifting links and each AND gate having another input for coupling to a source of a SHIFT not signal indicating when no CRC calculation is to be performed and only a shift of said CRC bits in said storage means (30) is to be performed, said shift being by one byte toward the predetermined byte position in said storage means (30).

7. An apparatus as defined in claim 1 further comprising a multiplexer (72) having a plurality of data inputs, each data input having a plurality of conductors coupled to the data outputs of one of said CRC bytes stored in said storage means (30), said multiplexer having one data output having the number of conductors equal to the number of bits in one byte of CRC bits, and said multiplexer having an input for receiving a byte select signal (74) indicating which CRC byte to select, and means for coupling the input of said multiplexer coupled to the outputs upon which the selected CRC byte resides to the said output bus (70).

8. An apparatus according to claim 1 wherein said CRC output bus is coupled to the highest order byte of CRC data stored in the storage means (30) and wherein the calculation means (84) includes signal generation means for simultaneously performing an exclusive-OR logical operation on each bit in said highest order byte of CRC bits stored in the storage means (30) with one bit of the raw input data to generate a first signal for each said CRC bit, and for simultaneously performing a logical AND operation between each result of the above defined exclusive-OR operations and a second signal so as to generate third resultant signals which are the AND logic functions between each of said first signals and said second signal, and means for calculating the CRC bit when said second signal is in a first predetermined logic state by performing an exclusive-OR logic operation between predetermined bits being shifted during predetermined shifts and the third resultant signals resulting from predetermined ones of said logical AND operations, and, when said second signal is in a second predetermined logic state, for only shifting said CRC bits toward the most significant CRC bit position in said storage means (30) without altering any CRC bit.

9. An apparatus as defined in claim 8 wherein the means for shifting is comprised of an array of shifting links (60) arranged into rows and columns with the first row having inputs coupled to said data outputs of said storage means (30) and having outputs where the output of each shifting link in the first row is coupled to the input of the next most significant shifting link in the next row except for the most significant shifting link in the first row which has its output coupled to the input of the least significant shifting link in the second row and wherein said second row of shifting links is coupled between said first and third rows in a like fashion as said first row was coupled between said storage means (30) and said second row and so on for each row, and wherein selected ones of said shifting links are exclusive-OR gates which exclusive-OR their data inputs with said resultant signals generated by said signal generation means (84).

10. An apparatus as defined in claim 9 wherein said signal generation means (84) is comprised of a plurality of exclusive-OR gates and a plurality of AND gates wherein said third resultant signal for said first row is generated by an exclusive-OR gate (62) having an input coupled to the data output of the most significant CRC bit stored in said storage means (30) and having an input for receiving the first bit of said raw input data which would arrive if the input data arrived in serial format, the output of said exclusive-OR gate being coupled to one input of an AND gate (66) which has another input for coupling to a source of said second signal, said AND gate (66) having an output coupled to an input of each exclusive-OR gate in said first row of said array of shifting links, and said third resultant signal for said second row being generated by a similar gating structure as is used to generate the third resultant signal for said first row except that the exclusive-OR gate (64) has one input coupled to the second most significant CRC bit data output from said storage means (30) and a second input for receiving the second bit of said raw data which would arrive if the raw input data were input in serial fashion, and so on for generation of the third resultant signals for all rows.

11. An apparatus as defined in claim 10 wherein said means for shifting (60) causes shifting of the CRC bits by the number of places equal to the number of bits in one byte when said second signal is in the logic zero state and said byte clock signal is asserted.

12. An apparatus according to claim 1 wherein the storage means (30) comprises a CRC checksum register having a plurality of memory cells arranged to store the plurality of bytes of CRC data, each said byte comprised of a plurality of CRC bits, and each memory cell having a data input, a data output and a byte clock input for coupling to said byte clock signal, each memory cell storing the data at said data input upon receipt of said byte clock signal at said byte clock input and presenting said stored data at said data output.

13. Apparatus according to claim 12 wherein the shifting means (60) comprises:

a plurality of data input exclusive-OR gates (62, 64), each having one input for coupling to one bit of the raw input data and each having one data input coupled to said data output of one of said memory cells coupled to said CRC data output bus (70), each data input exclusive-OR gate having a data output;

an AND gate (66, 68) for each said data input exclusive-OR gate, said AND gate having a data output and each said AND gate having one input coupled to said data output of one of said data input exclusive-OR gates and each said AND gate having another input for coupling to a SHIFT signal indicating when asserted that only a shift without any alteration of the CRC bits is to occur; and

a plurality of shifting links arranged into an array comprised of a number of rows equal to the number of said data input exclusive-OR gates and a number of columns equal to the number of bits in said CRC data and with each shifting link having an input and an output, said first row having its inputs coupled to the data outputs of said memory cells and said last row having its data outputs coupled to the data inputs of said memory cells, each row of shifting links comprised of a plurality of logical operators, one in each column, each logical operator performing a shift of the bit in that logical operator's column to the next higher order bit position in the next row, selected logical operators in each row also performing an exclusive-OR operation on their incoming bits with the output of said AND gate (66, 68) coupled to a data input of each exclusive-OR gate in the corresponding row, where the data input exclusive-OR gate (62) coupled to the most significant CRC bit corresponds to the first row and the data input exclusive-OR gate (64) coupled to the next most significant CRC bit corresponds to the second row and so on for each row.

14. A method for outputting a plurality of bytes of CRC check bits one byte at a time on an output bus (70) which is only wide enough to carry one byte at a time away from an apparatus for calculating CRC bits using a CRC checksum register (30) and a calculation means (84), the method comprising the step of calculating the CRC check bits using said calculation means (84) and storing them in said checksum register (30); and being characterised by comprising the further steps of:
outputting the highest order byte of CRC check bits on said output bus (70);
shifting the second most significant byte of CRC check bits into the position in said CRC checksum register formerly occupied by the highest order byte of CRC check bits while simultaneously shifting all lower order bytes one byte toward the most significant bit position:
outputting the second most significant byte of CRC check bits on said output bus; and
continuing said shifting and outputting steps until all bytes of CRC check bits have been output on said output bus.

15. The method of claim 14 wherein the calculation means used (84) is an array of rows of shifting links (60) having a number of rows equal to the number of bits in a byte and a number of columns equal to the number of CRC bits in said checksum register (30) and wherein each said row of shifting links shifts its input data one bit position toward the most significant bit position without altering said CRC data at predetermined times and supplies the shifted data to the next row except that the first row receives its data from the CRC checksum register and the last row outputs its data to the CRC checksum register and except that the most significant bit position shifting link in each row supplies its output to the least significant bit position shifting link of the next row and wherein selected shifting links are exclusive-OR gates and wherein the steps of shifting the bytes of CRC data one byte toward the most significant bit position includes the step of rendering each exclusive-OR gate in each row transparent by forcing a logic zero at one input of each exclusive-OR gate in each said row such that data from said CRC checksum register ripples unchanged through all rows of the shifting links and is re-input to the CRC checksum register on the next byte clock cycle shifted one byte toward the most significant bit.

Patentansprüche

1. CRC-Rechenvorrichtung mit einer Einrichtung (30) zum Speichern mehrerer CRC-Bytes, und einer Recheneinrichtung (84) zum Akzeptieren von Eingangsdaten von der Speichereinrichtung (30) und von Roh-Eingangsdaten, wobei die Recheneinrichtung (84) die Roh-Eingangsdaten durch eine vorbestimmte Binärzahl dividiert und den Rest der Division als CRC-Prüfbits in der Speichereinrichtung (30) speichert, gekennzeichnet durch
eine mit der Speichereinrichtung (30) und der Recheneinrichtung (84) verbundene Schiebeeinrichtung (60) zum Verschieben der CRC-Bytes in der Speichereinrichtung synchron mit einem Byte-Taktsignal um jeweils ein Byte auf einmal zu einer vorbestimmten Byte-Position hin durch Deaktivieren der Recheneinrichtung und Senden der aktuellen Inhalte der Speichereinrichtung unverändert durch die Recheneinrichtung bei gleichzeitigem Verschieben der CRC-Bytes und anschließendem Speichern der unveränderten, verschobenen CRC-Bits zurück in die Speichereinrichtung an Positionen, die um ein Byte zu der vorbestimmten Byte-Position hin verschoben sind, wobei die genannten Aktionen als Reaktion auf jede Aktivierung des Byte-Taktsignals einmal durchgeführt werden, und durch einen CRC-Ausgangsbus (70), der für jedes CRC-Bit in dem in der vorbestimmten Byte-Position gespeicherten CRC-Daten-Byte mit einem Leiter mit jedem jeweiligen Datenausgang der Speichereinrichtung (30) verbunden ist, wobei die CRC-Bytes über den Ausgangsbus (70) jeweils um ein Byte auf einmal zur Ausgabe verfügbar gemacht werden.

2. Vorrichtung nach Anspruch 1, bei der die Recheneinrichtung (84) synchron mit dem Byte-Takt Eingangsdaten akzeptiert, und zwar ein Byte auf einmal, und gleichzeitig unter Verwendung sämtlicher Bits des Eingangsdaten-Byte CRC-Bits errechnet.

3. Vorrichtung nach Anspruch 2, bei der die Recheneinrichtung (84) mehrere Reihen von Verschiebegliedern (60) enthält, wobei jedes Verschiebeglied einen Eingang und einen Ausgang aufweist, und wobei für jedes Eingangsdatenbit eine Reihe vorgesehen ist, und bei der jede Verschiebegliedreihe über ihre Eingänge mit den Ausgängen der Verschiebeglieder nächstniedrigster Signifikanz in der vorhergehenden Reihe verbunden ist, mit Ausnahme desjenigen Verschiebeglieds niedrigster Signifikanz in jeder Reihe, welches über seinen Eingang mit dem Ausgang des Verschiebe-

glieds höchster Signifikanz in der vorhergehenden Reihe verbunden ist, und mit Ausnahme der ersten Reihe, die über ihre Eingänge mit den Datenausgängen der Speichereinrichtung (30) so verbunden ist, als ob sie eine vorhergehende Reihe von Verschiebegliedern wäre, und bei der die letzte Reihe von Verschiebegliedern über ihre Datenausgänge mit den Dateneingängen der Speichereinrichtung (30) so verbunden ist, als ob sie eine vorhergehende Reihe von Verschiebegliedern wäre, und bei der vorbestimmte Verschiebeglieder Gatter sind, die eine logische Exklusiv-ODER-Operation zwischen ihren Eingangsbits und einem vorbestimmten Signal ausführen.

4. Vorrichtung nach Anspruch 3, bei der für jede Reihe der Verschiebeglieder ein vorbestimmtes Signal vorgesehen ist und jedes der vorbestimmten Signale durch ein Eingangs-Gatter erzeugt wird, das derjenigen bestimmten Reihe entspricht, die eine vorbestimmte logische Operation zwischen einem vorbestimmten der CRC-Bits und einem Daten-Bit von dem Eingangsdaten-Byte durchführt.

5. Vorrichtung nach Anspruch 4, bei der das der ersten Reihe entsprechende Eingangs-Gatter ein Gatter (62) ist, das eine logische Exklusiv-ODER-Operation zwischen dem in der Speichereinrichtung (30) gespeicherten Bit höchster Signifikanz und dem ersten Daten-Bit durchführt, welches eintreffen würde, wenn die Daten seriell eingegeben würden, und bei der das der zweiten Reihe entsprechende Eingangs-Gatter ein Gatter (64) ist, das eine Exklusiv-ODER-Operation zwischen dem in der Speichereinrichtung (30) gespeicherten CRC-Bit nächsthöchster Signifikanz und dem zweiten Bit durchführt, welches eintreffen würde, wenn die Eingangsdaten seriell eingegeben würden, und so fort für jede Reihe der Verschiebeglieder.

6. Vorrichtung nach Anspruch 5, bei der die Schiebeeinrichtung mehrere UND-Gatter (66,68) aufweist und dabei für jede Reihe der Verschiebeglieder jeweils eines vorgesehen ist, wobei jedes UND-Gatter über einen Eingang mit dem Ausgang des Exklusiv-ODER-Gatters für eine bestimmte Reihe verbunden ist und über seinen Ausgang mit einem der Eingänge der Exklusiv-ODER-Gatter in die bestimmte entsprechenden Reihe von Verschiebegliedern verbunden ist und jedes UND-Gatter einen weiteren Eingang zur Verbindung mit einer Quelle eines Nicht-SHIFT-Signals aufweist, welches angibt, wann keine CRC-Rechnung durchgeführt werden soll und lediglich eine Verschiebung der CRC-Bits in der Speichereinrichtung (30) durchgeführt werden soll, wobei die Verschiebung um ein Byte auf die vorbestimmte Byte-Position in der Speichereinrichtung (30) hin erfolgt.

7. Vorrichtung nach Anspruch 1, ferner mit einem Multiplexer (72) mit mehreren Dateneingängen, wobei jeder Dateneingang mehrere Leiter aufweist, die mit den Datenausgängen eines der in der Speichereinrichtung (30) gespeicherten CRC-Bytes verbunden ist, wobei der Multiplexer einen Datenausgang aufweist, bei dem die Anzahl der Leiter der Anzahl von Bits in einem Byte von CRC-Bits gleich ist, und der Multiplexer einen Eingang zum Empfangen eines Bytewählsignals (74) aufweist, das angibt, welches CRC-Byte gewählt werden soll, und mit einer Einrichtung, die den Eingang des Multiplexers, der mit den Ausgängen verbunden ist, an welchen das gewählte CRC-Byte liegt, mit dem Ausgangsbus (70) verbindet.

8. Vorrichtung nach Anspruch 1, bei der der CRC-Ausgangsbus mit dem ranghöchsten Byte der in der Speichereinrichtung (30) gespeicherten CRC-Daten verbunden ist und bei der die Recheneinrichtung (84) eine Signalerzeugungseinrichtung enthält, die gleichzeitig eine logische Exklusiv-ODER-Operation für jedes Bit in dem ranghöchsten Byte der in der Speichereinrichtung (30) gespeicherten CRC-Bits mit einem Bit der Roh-Eingangsdaten durchführt, um ein erstes Signal für jedes CRC-Bit zu erzeugen, und die gleichzeitig eine logische UND-Operation zwischen jedem Ergebnis der vorstehend definierten Exklusiv-ODER-Operationen und einem zweiten Signal durchführt, um dritte Ergebnis-Signale zu erzeugen, die die logischen UND-Funktionen zwischen jedem der ersten Signale und dem zweiten Signal sind, und bei der eine Einrichtung vorgesehen ist, die, wenn sich das zweite Signal in einem ersten vorbestimmten Logik-Zustand befindet, das CRC-Bit errechnet, indem sie eine logische Exklusiv-ODER-Operation zwischen vorbestimmten Bits, die während vorbestimmter Verschiebungen verschoben werden, und den aus den vorbestimmten der logischen UND-Operationen resultierenden dritten Ergebnis-Signalen durchführt, und die, wenn sich das zweite Signal in einem zweiten vorbestimmten Logik-Zustand befindet, die CRC-Bits lediglich zu der in der Speichereinrichtung (30) befindlichen CRC-Bit-Position höchster Signifikanz hin verschiebt, ohne irgendein CRC-Bit zu ändern.

9. Vorrichtung nach Anspruch 8, bei der die zum Verschieben vorgesehene Einrichtung ein Array von Verschiebegliedern (60) aufweist, die in Reihen und Spalten angeordnet sind, wobei die erste Reihe Eingänge, die mit den Datenausgängen der Speichereinrichtung (30) verbunden sind, und Ausgänge aufweist, wobei der Ausgang jedes Verschiebegliedes in der ersten Reihe mit dem Eingang des Verschiebegliedes nächsthöherer Signifikanz in der nächsten Reihe verbunden ist, mit Ausnahme des Verschiebegliedes höchster Signifikanz in der ersten Reihe, das über seinen Ausgang mit dem Eingang des Verschiebegliedes niedrigster Signifikanz in der zweiten Reihe verbunden ist, und bei der die zweite Reihe von Verschiebegliedern in der gleichen Weise, in der die erste Reihe zwischen die Speichereinrichtung (30) und die zweite Reihe verbunden war, zwischen die erste und die dritte Reihe geschaltet ist usw. für jede Reihe, und bei der gewählte der Verschiebeglieder Exklusiv-ODER-Gatter sind, die ihre Dateneingänge mit den von der Signalerzeugungseinrichtung (84) erzeugten Ergebnis-Signalen einer Exklusiv-ODER-Operation unterziehen.

10. Vorrichtung nach Anspruch 9, bei der die Signalerzeugungseinrichtung (84) mehrere Exklusiv-ODER-Gatter und mehrere UND-Gatter aufweist, wobei das dritte Ergebnis-Signal für die erste Reihe von einem Exklusiv-ODER-Gatter (62) erzeugt wird, das einen Eingang, der mit dem Datenausgang des in der Speichereinrichtung (30) gespeicherten CRC-Bits höchster Signifikanz verbunden ist, und einen Eingang zum Empfangen des ersten Bits der Roh-Eingangsdaten aufweist, welches eintreffen würde, wenn die Eingangsdaten seriell eintreffen würden, wobei der Ausgang des Exklusiv-ODER-Gatters mit einem Eingang eines UND-Gatters (66) verbunden ist, das einen weiteren Eingang zur Verbindung mit einer Quelle des zweiten Signals aufweist, wobei das UND-Gatter (66) einen Ausgang aufweist, der mit einem Eingang jedes Exklusiv-ODER-Gatters in der ersten Reihe des Array von Verschiebegliedern verbunden ist, und wobei das dritte Ergebnis-Signal für die zweite Reihe von einer Gatterstruktur erzeugt wird, die der zum Erzeugen des dritten Ergebnis-Signals für die erste Reihe verwendeten mit Ausnahme der Tatsache gleicht, daß das Exklusiv-ODER-Gatter (64) einen Eingang, der mit dem von der Speichereinrichtung (30) kommenden CRC-Bit-Datenausgang zweithöchster Signifikanz verbunden ist, und einen zweiten Eingang zum Empfangen des zweiten Bits der Roh-Daten aufweist, welches eintreffen würde, wenn die Roh-Eingangsdaten seriell eingegeben würden, und so fort zur Erzeugung der dritten Ergebnis-Signale für sämtliche Reihen.

11. Vorrichtung nach Anspruch 10, bei der die zum Verschieben vorgesehene Einrichtung (60) eine Verschiebung der CRC-Bits um die der Anzahl von Bits in einem Byte gleichende Anzahl von Stellen bewirkt, wenn das zweite Signal sich in einem logischen Null-Zustand befindet und das Byte-Takt-Signal vorliegt.

12. Vorrichtung nach Anspruch 1, bei der die Speichereinrichtung (30) ein CRC-Prüfsummenregister mit mehreren Speicherzellen aufweist, die zum Speichern der mehreren Bytes von CRC-Daten ausgebildet sind, wobei jedes Byte mehrere CRC-Bits aufweist und jede Speicherzelle einen Dateneingang, einen Datenausgang und einen Byte-Takt-Eingang zur Verbindung mit dem Byte-Takt-Signal aufweist, wobei jede Speicherzelle bei Empfang des Byte-Takt-Signals an dem Byte-Takt-Eingang die Daten an dem Dateneingang speichert und die gespeicherten Daten an den Datenausgang verfügbar macht.

13. Vorrichtung nach Anspruch 12, bei der die Schiebeeinrichtung (60) aufweist:

mehrere Dateneingabe-Exklusiv-ODER-Gatter (62, 64), von denen jedes einen Eingang zur Verbindung mit einem Bit der Roh-Eingangsdaten aufweist und von denen jedes einen Dateneingang aufweist, der mit dem Datenausgang einer der mit dem CRC-Datenausgangsbus (70) verbundenen Speicherzellen verbunden ist, wobei jedes der Dateneingabe-Exklusiv-ODER-Gatter einen Datenausgang aufweist;

ein UND-Gatter (66,68) für jedes der Dateneingabe-Exklusiv-ODER-Gatter, wobei das UND-Gatter einen Datenausgang aufweist und jedes der UND-Gatter einen Eingang aufweist, der mit dem Datenausgang eines der Dateneingabe-Exklusiv-ODER-Gatter verbunden ist, und jedes der UND-Gatter einen Eingang zur Verbindung mit einem SHIFT-Signal aufweist, das, wenn es vorliegt, angibt, daß lediglich eine Verschiebung ohne jegliche Änderung der CRC-Bits erfolgen soll; und

mehrere Verschiebeglieder, die zu einem Array angeordnet sind, das eine der Anzahl der Dateneingabe-Exklusiv-ODER-Gatter gleichende Anzahl von Reihen und eine der Anzahl der Bits in den CRC-Daten gl ichende Anzahl von Spalten aufweist, und wobei jedes Verschiebeglied einen Eingang und einen Aus-

gang aufweist, wobei die erste Reihe über ihre Eingänge mit den Daten-Ausgängen der Speicherzellen verbunden ist und die letzte Reihe über ihre Daten-Ausgänge mit den Daten-Eingängen der Speicherzellen verbunden ist, wobei jede Reihe von Verschiebegliedern mehrere Logik-Operatoren aufweist, von denen jeweils einer in jeder Spalte vorgesehen ist, wobei jeder Logik-Operator eine Verschiebung des Bits in dieser Spalte des Logik-Operators auf die nächsthöherrangige Bitposition in der nächsten Reihe durchführt, wobei gewählte Logik-Operatoren in jeder Reihe auch die bei ihnen eingehenden Bits einer Exklusiv-ODER-Operation mit dem Ausgang des UND-Gatters (66,68) unterziehen, das mit einem Daten-Eingang jedes Exklusiv-ODER-Gatters in der entsprechenden Reihe verbunden ist, wobei das mit dem CRC-Bit höchster Signifikanz gekoppelte Dateneingabe-Exklusiv-ODER-Gatter (62) der ersten Reihe entspricht und das mit dem CRC-Bit nächsthöherer Signifikanz gekoppelte Dateneingabe-Exklusiv-ODER-Gatter (64) der zweiten Reihe entspricht, und so fort für jede Reihe.

14. Verfahren zum Ausgeben mehrerer Bytes von CRC-Prüf-Bits auf einen Ausgangsbus (70), und zwar ein Byte auf einmal, wobei der Ausgangsbus (70) lediglich breit genug ist, um ein Bit auf einmal von einer Vorrichtung, die zum Errechnen von CRC-Bits ein CRC-Prüfsummenregister (30) und eine Recheneinrichtung (84) verwendet, wegzuführen, wobei das Verfahren den Schritt des Errechnens der CRC-Prüf-Bits mittels der Recheneinrichtung (84) und des Speicherns derselben in dem Prüfsummenregister (30) umfaßt und gekennzeichnet ist durch die folgenden Schritte:

Ausgeben des höchstrangigen Bytes der CRC-Prüf-Bits auf dem Ausgangsbus (70);

Verschieben des Bytes zweithöchster Signifikanz der CRC-Prüf-Bits auf die Position in dem CRC-Prüfsummenregister, die zuvor von dem höchstrangigen Byte der CRC-Prüf-Bits eingenommen wurde, während gleichzeitig sämtliche niedrigerrangigen Bytes um ein Byte zu der Bitposition höchster Signifikanz hin verschoben werden;

Ausgeben des Bytes zweithöchster Signifikanz der CRC-Prüf-Bits auf den Ausgangsbus; und

Fortsetzen der Schiebe- und Ausgabeschritte, bis samtliche Bytes der CRC-Pruf-Bits auf den Ausgangsbus ausgegeben worden sind.

15. Verfahren nach Anspruch 14, bei dem die verwendete Recheneinrichtung (84) ein Array von Reihen von Verschiebegliedern (60) mit einer der Anzahl von Bits in einem Byte gleichenden Anzahl von Reihen ist und mit einer der Anzahl von CRC-Bits in dem Prüfsummenregister (30) gleichenden Anzahl von Spalten ist, und bei dem jede der Reihen von Verschiebegliedern zu vorbestimmten Zeiten ihre Eingangsdaten um eine Bit-Position zu der Bit-Position höchster Signifikanz hin verschiebt, ohne die CRC-Daten zu verändern, und die verschobenen Daten der nächsten Reihe mit Ausnahme der Tatsache zuführt, daß die erste Reihe ihre Daten von dem CRC-Prüfsummenregister empfängt und die letzte Reihe ihre Daten an das CRC-Prüfsummenregister ausgibt, und ferner mit Ausnahme der Tatsache zuführt, daß das Verschiebeglied für die Bitposition höchster Signifikanz in jeder Reihe sein Ausgangssignal dem Verschiebeglied für die Bitposition niedrigster Signifikanz der nächsten Reihe zuführt, und bei dem ausgewählte Verschiebeglieder Exklusiv-ODER-Gatter sind, und bei dem der Schritt des Verschiebens der Bytes der CRC-Daten um ein Byte zu der Bit-Position höchster Signifikanz hin einen Schritt enthält, bei dem jedes Exklusiv-ODER-Gatter in jeder Reihe durch zwangsweises Anlegen einer logischen Null an einen Eingang jedes Exklusiv-ODER-Gatter in jeder solchen Reihe transparent gemacht wird, derart, daß ein Datenwert von dem CRC-Prüfsummenregister unverändert sämtliche Reihen der Verschiebeglieder durchläuft und in dem nächsten Byte-Takt-Zyklus um ein Byte zu dem Bit höchster Signifikanz versetzt wieder in das CRC-Prüfsummenregister eingegeben wird.

Revendications

1. Dispositif de calcul de CRC (contrôle de redondance cyclique) comprenant des moyens de mémoire (30) pour une pluralité d'octets CRC, et des moyens de calcul (84) pour accepter des données d'entrée provenant des moyens de mémoire (30) et des données d'entrée brutes, les moyens de calcul (84) divisant les données d'entrée brutes par un nombre binaire prédéterminé et mémorisant le reste de la division dans les moyens de mémoire (30) sous la forme de bits de contrôle CRC, caractérisé par des moyens de décalage (60) couplés au moyens de mémoire (30) et aux moyens de calcul (84) pour décaler les octets CRC en raison d'un octet à la fois, dans les moyens de mémoire en direction d'une position d'octet prédéterminée en synchronisme

avec un signal d'horloge d'octet par désactivation des moyens de calcul et envoi du contenu actuel des moyens de mémoire dans les moyens de calcul sans modification, tout en décalant simultanément les octets CRC, puis en mémorisant à nouveau les bits CRC inchangés et décalés dans les moyens de mémoire dans des positions qui sont décalées d'un octet en direction de la position d'octet prédéterminée, les actions précédentes étant exécutées une fois en réponse à chaque activation du signal d'horloge d'octet, et un bus de sortie de CRC (70) comportant un conducteur couplé à chaque sortie respective de données des moyens de mémoire (30) pour chaque bit CRC dans l'octet de données CRC mémorisé dans la position d'octet prédéterminée, ce qui rend les octets CRC accessibles pour leur délivrance, en raison d'un octet à la fois, par l'intermédiaire dudit bus de sortie (70).

2. Dispositif selon la revendication 1, dans lequel des moyens de calcul (84) acceptent des données d'entrée à raison d'un octet à la fois, en synchronisme avec ledit signal d'horloge d'octet, et calculent simultanément des bits CRC utilisant tous les bits de l'octet de données d'entrée.

3. Dispositif selon la revendication 2, dans lequel lesdits moyens de calcul (84) comprennent une pluralité de rangées de liaisons de décalage (60), dont chacune possède une entrée et une sortie, l'une desdites rangée étant prévue pour chaque bit de donnée d'entrée, et dans lequel les entrées de chaque rangée de liaisons de décalage sont couplées aux sorties des liaisons de décalage suivantes les moins importantes dans la ligne précédente, hormis pour la liaison de décalage la moins importante de chaque ligne, dont l'entrée est couplée à la sortie de la liaison de décalage la plus importante dans la ligne précédente et hormis pour la première ligne, dont les entrées sont couplées auxdites sorties de données desdits moyens de mémoire (30) comme s'il s'agissait d'une ligne précédente de liaisons de décalage, et dans lequel les sorties de données de la dernière ligne de liaisons de décalage sont couplées auxdites entrées de données desdits moyens de mémoire (30) comme s'il s'agissait d'une ligne précédente de liaisons de décalage, et dans lequel des liaisons de décalage prédéterminées sont des portes, qui exécutent une opération logique OU-Exclusif entre leurs bits d'entrée et un signal prédéterminé.

4. Dispositif selon la revendication 3, dans lequel il existe un signal prédéterminé pour chacune desdites lignes desdites liaisons de décalage, et chacun desdits signaux prédéterminés est produit par une porte d'entrée qui correspond à cette ligne particulière, qui exécute une opération logique prédéterminée entre l'un prédéterminé desdits bits CRC et un bit de donnée délivré par lesdits octets d'entrée de données.

5. Dispositif selon la revendication 4, dans lequel la porte d'entrée, qui correspond à la première ligne, est une porte (62) qui exécute une opération logique OU-Exclusif entre le bit CRC de poids le plus élevé mémorisé dans les moyens de mémoire (30) et le premier bit de donnée, qui apparaîtrait si les données étaient introduites en série, et selon lequel la porte d'entrée, qui correspond à la seconde ligne, est une porte (64) exécute une opération OU-Exclusif entre le bit CRC de poids le plus faible immédiatement suivant, mémorisé dans les moyens de mémoire (30), et le second bit, qui sinon apparaîtrait si lesdites données d'entrée étaient introduites en parallèle, et ainsi de suite pour chaque ligne desdites liaisons de décalage.

6. Dispositif selon la revendication 5, dans lequel lesdits moyens de décalage sont constitués par une pluralité de portes ET 66-68, une pour chaque ligne desdites liaisons de décalage, une entrée de chaque porte ET étant couplée à la sortie de la porte OU-Exclusif pour une ligne particulière tandis que sa sortie est couplée à l'une des entrées des portes OU-Exclusif dans la ligne particulière correspondante de liaisons de décalage, et chaque porte ET possédant une autre entrée pour son raccordement et une source d'un signal de NON-Décalage indiquant le moment où aucun calcul CRC ne doit être exécuté et où seulement un décalage desdits bits CRC dans lesdits moyens de mémoire (30) doit être effectué, ledit décalage s'effectuant au moyen d'un seul octet en direction de la position d'octet prédéterminée dans lesdits moyens de mémoire (30).

7. Dispositif selon la revendication 1, comprenant en outre un multiplexeur (72) possédant une pluralité d'entrées de données, dont chacune possède une pluralité de conducteurs couplés aux sorties de données de l'un desdits octets CRC mémorisés dans lesdits moyens de mémoire (30), ledit multiplexeur comportant une sortie de données comprenant un nombre de conducteurs égal au nombre de bits dans un octet de bits CRC, et ledit multiplexeur possédant une entrée servant à recevoir un signal

de sélection d'octet (74) indiquant quel octet CRC doit être sélectionné, et des moyens pour coupler l'entrée dudit multiplexeur, couplé aux sorties auxquelles l'octet sélectionné est présent, audit bus de sortie (70).

8. Dispositif selon la revendication 1, dans lequel ledit bus de sortie CRC est couplé à l'octet de rang maximum des données CRC mémorisées dans la mémoire (30), et dans lequel les moyens de calcul (84) comprennent des moyens de production de signaux servant à exécuter simultanément une opération logique OU-Exclusif sur chaque bit présent dans ledit octet de rang maximum de bits CRC mémorisés dans les moyens de mémoire (30) avec un bit des données d'entrée brut pour produire un premier signal pour chacun desdits bits CRC, et pour exécuter simultanément une opération logique ET entre chaque résultat des opérations OU-Exclusif définies ci-dessus et un second signal afin de produire des troisièmes signaux résultants, qui sont des combinaisons logiques ET entre chacun desdits premiers signaux et ledit second signal, et des moyens pour calculer le bit CRC lorsque ledit second signal est dans un premier état logique prédéterminé moyennant l'exécution d'une opération logique OU-Exclusif entre des bits prédéterminés transférés pendant des décalages prédéterminés et les troisièmes signaux résultants, qui proviennent de certaines prédéterminées desdites opérations logiques ET, et, lorsque le second signal de commande est dans un second état logique prédéterminé, pour décaler uniquement lesdits bits CRC en direction de la position de bit CRC la plus importante dans lesdits moyens de mémoire (30) sans modifier un quelconque bit CRC.

9. Dispositif selon la revendication 8, dans lequel les moyens de décalage sont constitués par un réseau de liaisons de décalage (60) disposées suivant des lignes et des colonnes parmi lesquelles la première ligne possède des entrées couplées auxdites sorties de données desdits moyens de mémoire (30) et comporte des sorties, la sortie de chaque liaison de décalage dans la première ligne étant couplée à l'entrée de la liaison de décalage la plus importante immédiatement suivante dans la ligne suivante, hormis pour la liaison de décalage la plus importante dans la première ligne, dont l'entrée est couplée à l'entrée de la liaison de décalage la moins importante dans la seconde ligne, et dans lequel ladite seconde ligne de liaisons de décalage est branchée entre lesdites première et troisième lignes de la même manière que si ladite première ligne était branchée entre lesdits moyens de mémoire (30) et ladite seconde ligne et ainsi de suite pour chaque ligne, et dans lequel des liaisons sélectionnées parmi lesdites liaisons de transfert sont des portes OU-Exclusif, qui réalisent la combinaison OU-Exclusif de leurs données d'entrée avec lesdits signaux résultants produits par lesdits moyens de production de signaux (84).

10. Dispositif selon la revendication 9, dans lequel lesdits moyens de production de signaux (84) sont constitués par une pluralité de portes OU-Exclusif et par une pluralité de portes ET, et dans lequel ledit troisième signal résultant pour ladite première ligne est produit par une porte OU-Exclusif (62), comportant une entrée couplée à la sortie de données du bit CRC de poids le plus important mémorisé dans lesdits moyens de mémoire (30) et possédant une entrée servant à recevoir le premier bit desdites données d'entrée brutes, qui se présenterait si les données d'entrée arrivaient selon le format en série, le signal de sortie de ladite porte OU-Exclusif étant couplé à une entrée d'une porte ET (66), qui comporte une autre entrée pour son raccordement à une source dudit second signal, ladite porte ET (66) possédant une sortie couplée à une entrée de chaque porte OU-Exclusif présente dans ladite première ligne dudit réseau de liaisons de transfert, et ledit troisième signal résultant pour ladite seconde ligne étant produit par une structure similaire de commande de transmission tel qu'elle est utilisée pour produire le troisième signal résultant pour ladite première ligne, hormis que la porte OU-Exclusif (64) possède une entrée couplée à la sortie de données du second bit CRC de poids le plus important à partir desdits moyens de mémoire (30), et une seconde entrée servant à recevoir le second bit desdites données brutes, qui apparaîtrait si les données d'entrée brutes étaient introduites en série, et ainsi de suite pour la production des troisièmes signaux résultants pour toutes les lignes.

11. Dispositif selon la revendication 10, dans lequel lesdits moyens de décalage (60) provoquent le décalage des bits CRC sur un nombre d'emplacements égal au nombre de bits dans un octet lorsque ledit second signal est dans l'état logique zéro et que ledit signal d'horloge d'octet est activé.

12. Dispositif selon la revendication 1, dans lequel les moyens de mémoire (30) comprennent un

registre de total de contrôl CRC possédant une pluralité de cellules de mémoire disposées de manière à mémoriser la pluralité d'octets de données CRC, chacun desdits octets étant constitué par une pluralité de bits CRC, et chaque cellule de mémoire possédant une entrée de données, une sortie de données et une entrée du signal d'horloge pour l'application dudit signal d'horloge d'octet, chaque cellule de mémoire mémorisant les données au niveau de ladite entrée de données lors de la réception dudit signal d'horloge d'octet au niveau de ladite entrée du signal d'horloge d'octet et présentant lesdites données mémorisées au niveau de ladite sortie de données.

13. Dispositif selon la revendication 12, dans lequel les moyens de décalage (60) comprennent :

une pluralité de portes OU-Exclusif d'entrées de données (62,64), dont chacune possède une entrée prévue pour son couplage à un bit des données d'entrée brutes et dont chacune possède une entrée de données couplée à ladite sortie de données de l'une desdites cellules de mémoire couplées audit bus de sortie de données CRC (70), chaque porte OU-Exclusif d'entrée de données possédant une sortie de données;

une porte ET (66,68) pour chacune desdites portes OU-Exclusif d'entrée de données, ladite porte ET possédant une sortie de données et chacune desdites portes ET possédant une entrée couplée à ladite sortie de données de l'une desdites portes OU-Exclusif d'entrée de données, et chacune desdites portes ET possédant une autre entrée destinée à recevoir un signal DECALAGE indiquant, lorsqu'il est activé, que seul un décalage sans aucune modification des bits CRC doit être exécuté; et

une pluralité de liaisons de décalage disposées selon un réseau constitué par un nombre de lignes égal au nombre desdites portes OU-Exclusif d'entrée de données et un nombre de colonnes égal au nombre de bits dans lesdites données CRC, chaque liaison de décalage possédant une entrée et une sortie, les entrées de ladite première ligne étant couplées aux sorties de données desdites cellules de mémoire, et les sorties de données de ladite dernière ligne étant couplées aux entrées de données desdites cellules de mémoire, chaque ligne de liaisons de décalage étant constituée par une pluralité d'opérateurs logiques, un dans chaque colonne, chaque opérateur logique exécutant un transfert du bit dans cette colonne de l'opérateur logique jusqu'à la position binaire de rang immédiatement supérieur dans la ligne suivante, des opérateurs logiques sélectionnés dans chaque ligne appliquant une opération OU-Exclusif à leurs bits d'arrivée, tandis que la sortie de ladite porte ET (66,68) est couplée à une entrée de données de chaque porte OU-Exclusif dans la ligne correspondante, et dans lequel la porte OU-Exclusif d'entrée de données (62) couplée au bit CRC de poids maximum correspond à la première ligne, et la porte OU-Exclusif d'entrée de données (64) couplée au bit CRC de poids maximum immédiatement suivant correspond à la seconde ligne, ainsi de suite pour chaque ligne.

14. Procédé pour délivrer une pluralité d'octets de bits de contrôle CRC, en raison d'un octet à la fois, à un bus de sortie (70), qui est d'une largeur juste suffisante pour véhiculer un octet à la fois à partir d'un dispositif pour calculer des bits CRC en utilisant un registre de total de contrôle CRC (30) et des moyens de calcul (84), le procédé comprenant les étapes de calcul des bits de contrôle CRC moyennant l'utilisation desdits moyens de calcul (84) et de mémorisation de ces bits dans ledit registre de total de contrôle (30); et étant caractérisé en ce qu'il comprend les étapes suivantes :

délivrance de l'octet de rang maximum des bits de contrôle CRC audit bus de sortie (70);

transfert du second octet le plus important des bits de contrôle CRC dans la position dudit registre de total de contrôle CRC occupé antérieurement par l'octet de rang maximum des bits de contrôle CRC, avec un décalage simultané de tous les octets de rang inférieur, sur un bit, en direction de la position binaire la plus élevée;

délivrance du second octet de poids maximum de bits de contrôle CRC dans ledit bus de sortie; et

poursuite desdites étapes de décalage et de délivrance jusqu'à ce que tous les octets formés de bits de contrôle CRC aient été délivrés dans ledit bus de sortie.

15. Procédé selon la revendication 14, selon lequel les moyens de calcul utilisés (84) sont formés par un réseau de lignes de liaisons de décalage (60) possédant un nombre de lignes égal au nombre de bits dans un octet et un nombre de colonnes égal au nombre de bits CRC dans ledit registre de total de contrôle (30), et selon lequel chacune desdites lignes des liaisons de décalage décalent ses données d'entrée, d'une position binaire, en direction du bit de poids maximum sans modifier les données CRC à

des intervalles de temps prédéterminés, et envoie les données transférées, à la ligne suivante, hormis que la première ligne reçoit ses données de la part du registre de total de contrôle CRC et que la dernière ligne délivre ses données au registre de total de contrôle CRC et hormis que la liaison de décalage de la position binaire la plus élevée dans chaque ligne délivre son signal de sortie à la liaison de décalage de la position binaire la plus basse de la ligne suivante, et selon lequel des liaisons de décalage sélectionnées sont des portes OU-Exclusif, et les étapes de décalage des octets de données CRC à raison d'un octet en direction de la position binaire la plus élevée incluent l'étape consistant à rendre passante chaque porte OU-Exclusif dans chaque ligne, par application forcée d'un zéro logique à une entrée de chaque porte OU-Exclusif dans chacune desdites lignes de telle sorte que des données provenant desdits registres de total de contrôle CRC sont transmises d'une manière inchangée dans toutes les lignes de liaison de décalage et sont réintroduites dans le registre de total de contrôle CRC lors du cycle suivant d'horloge d'octet, décalé d'un octet en direction du bit de poids maximum.

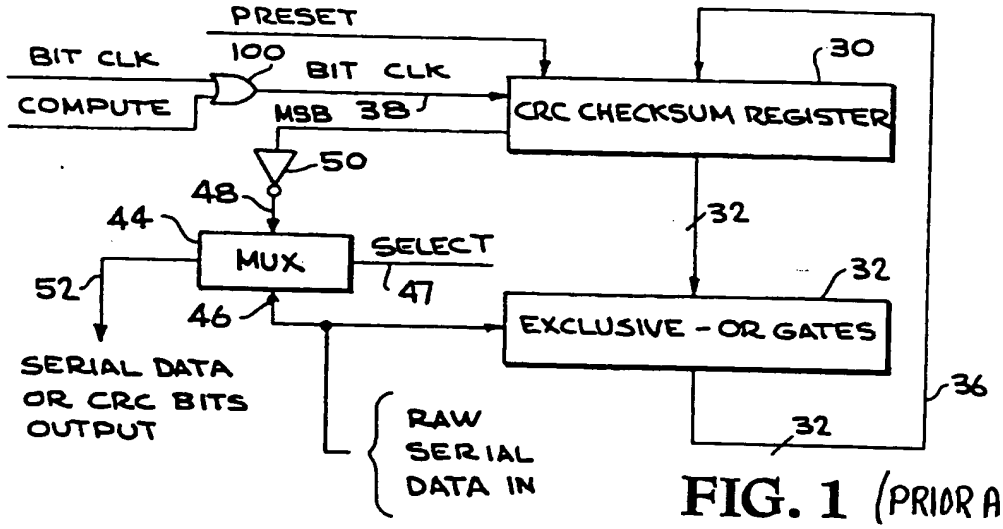


FIG. 1 (PRIOR ART)

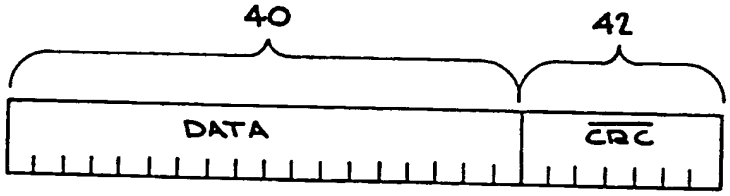


FIG. 3
(PRIOR ART)

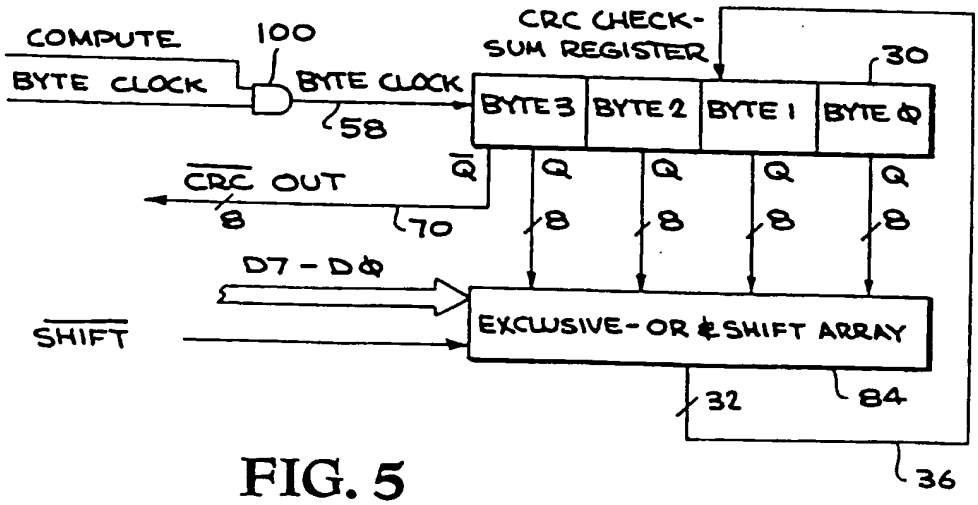


FIG. 5

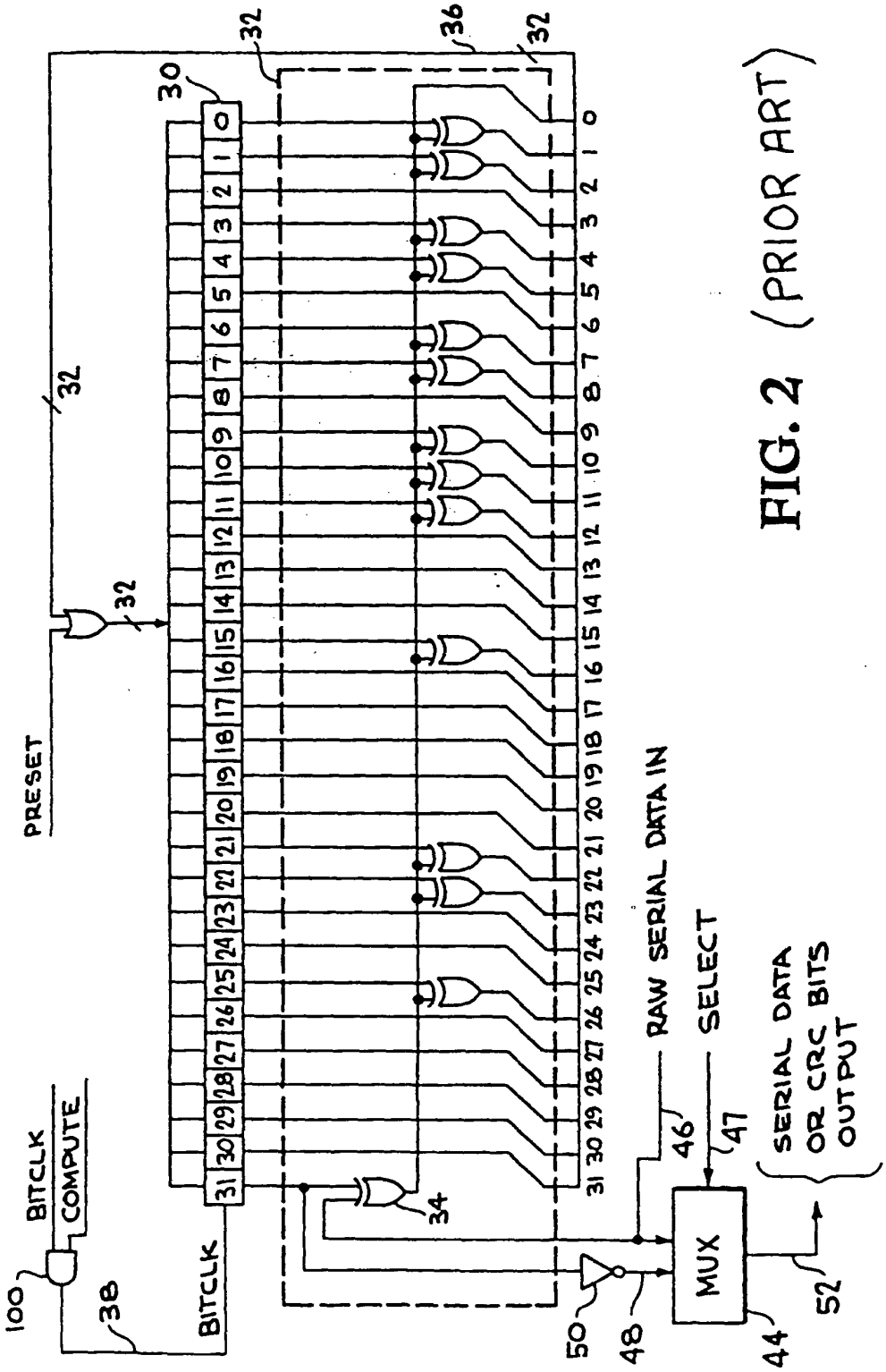
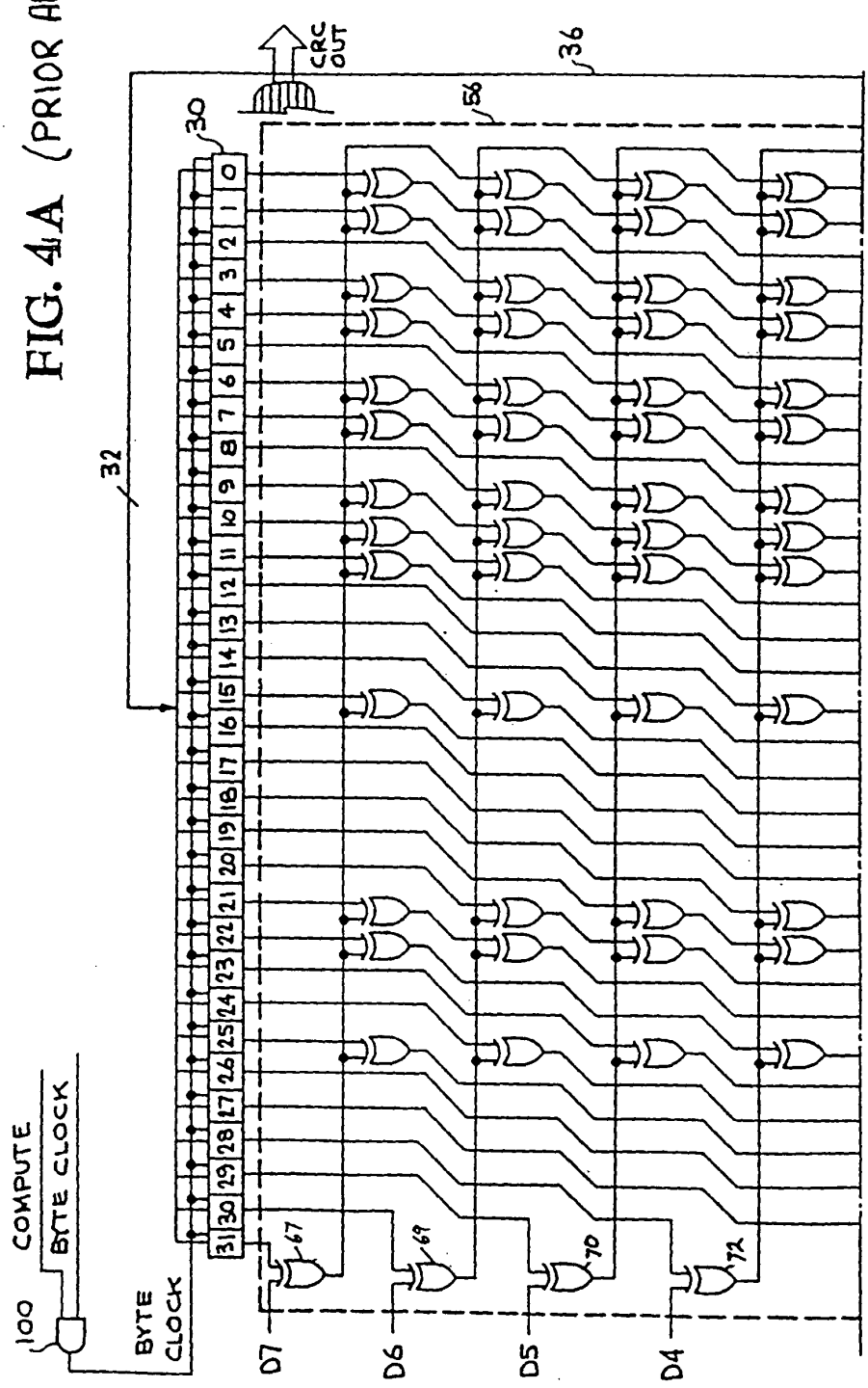


FIG. 2 (PRIOR ART)

FIG. 4A (PRIOR ART)



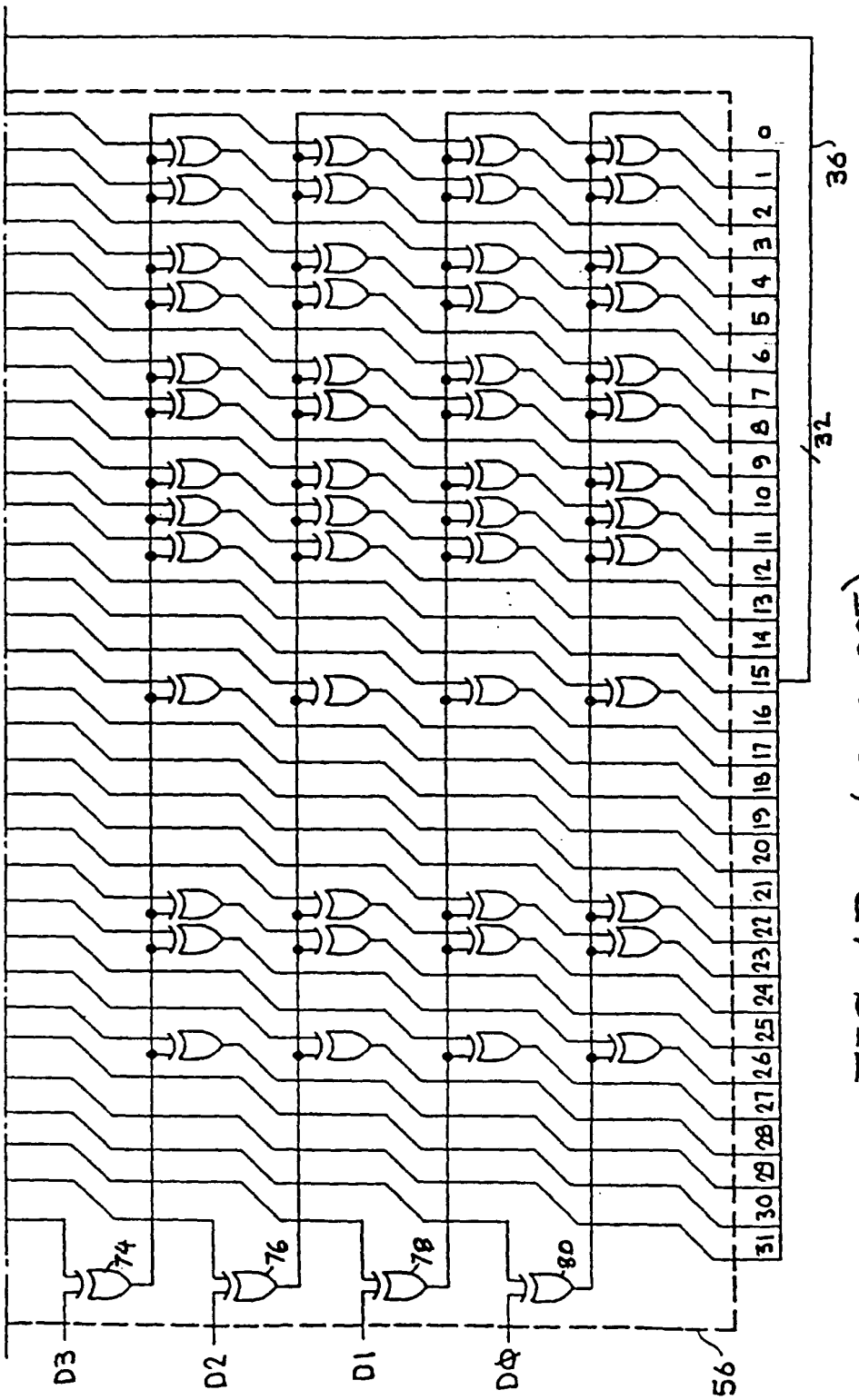


FIG. 4B (PRIOR ART)

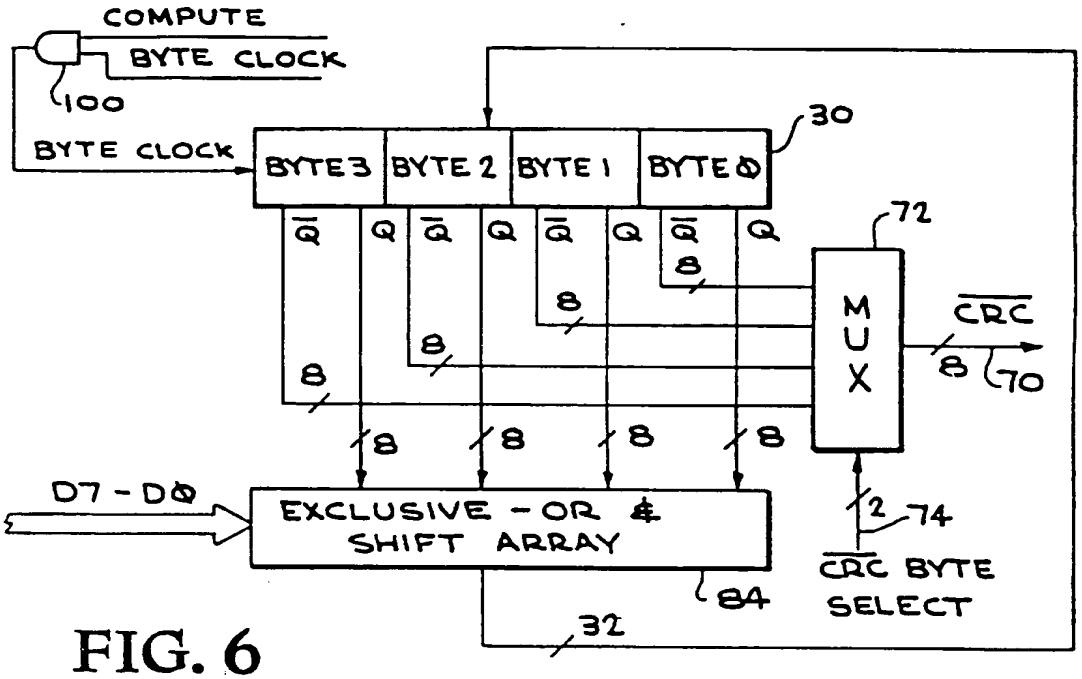

COMPUTE
BYTE CLOCK
100
BYTE CLOCK
BYTE 3
BYTE 2
BYTE 1
BYTE 0
30
72
MUX
CRC
8
70
D7 - D0
EXCLUSIVE - OR & SHIFT ARRAY
2
74
CRC BYTE SELECT
84
32

FIG. 6

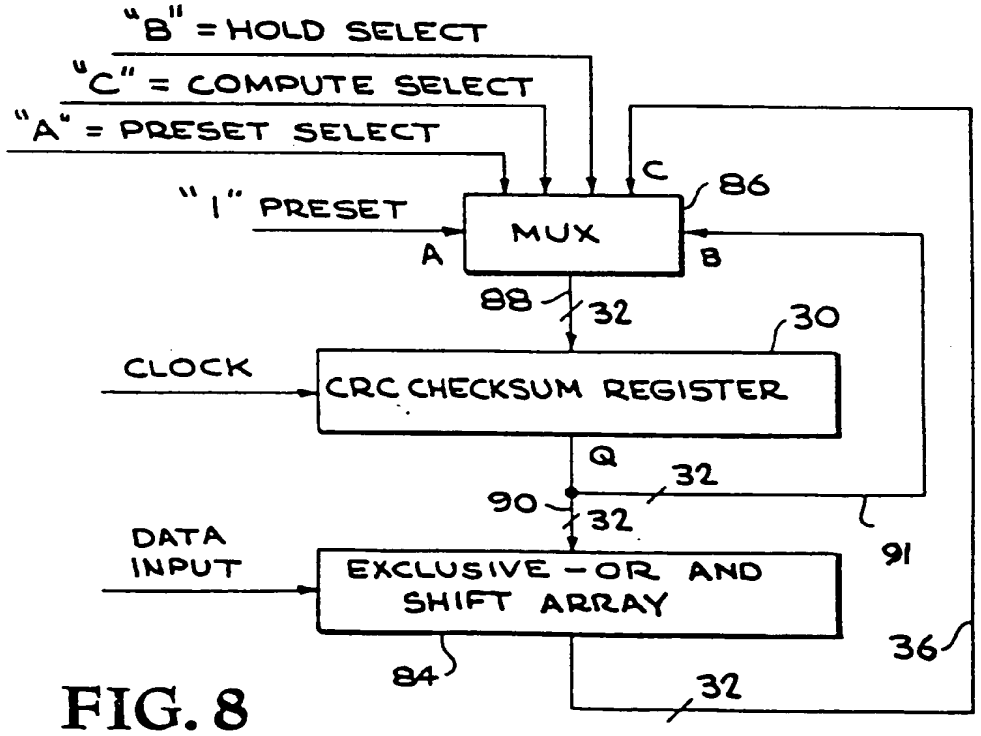
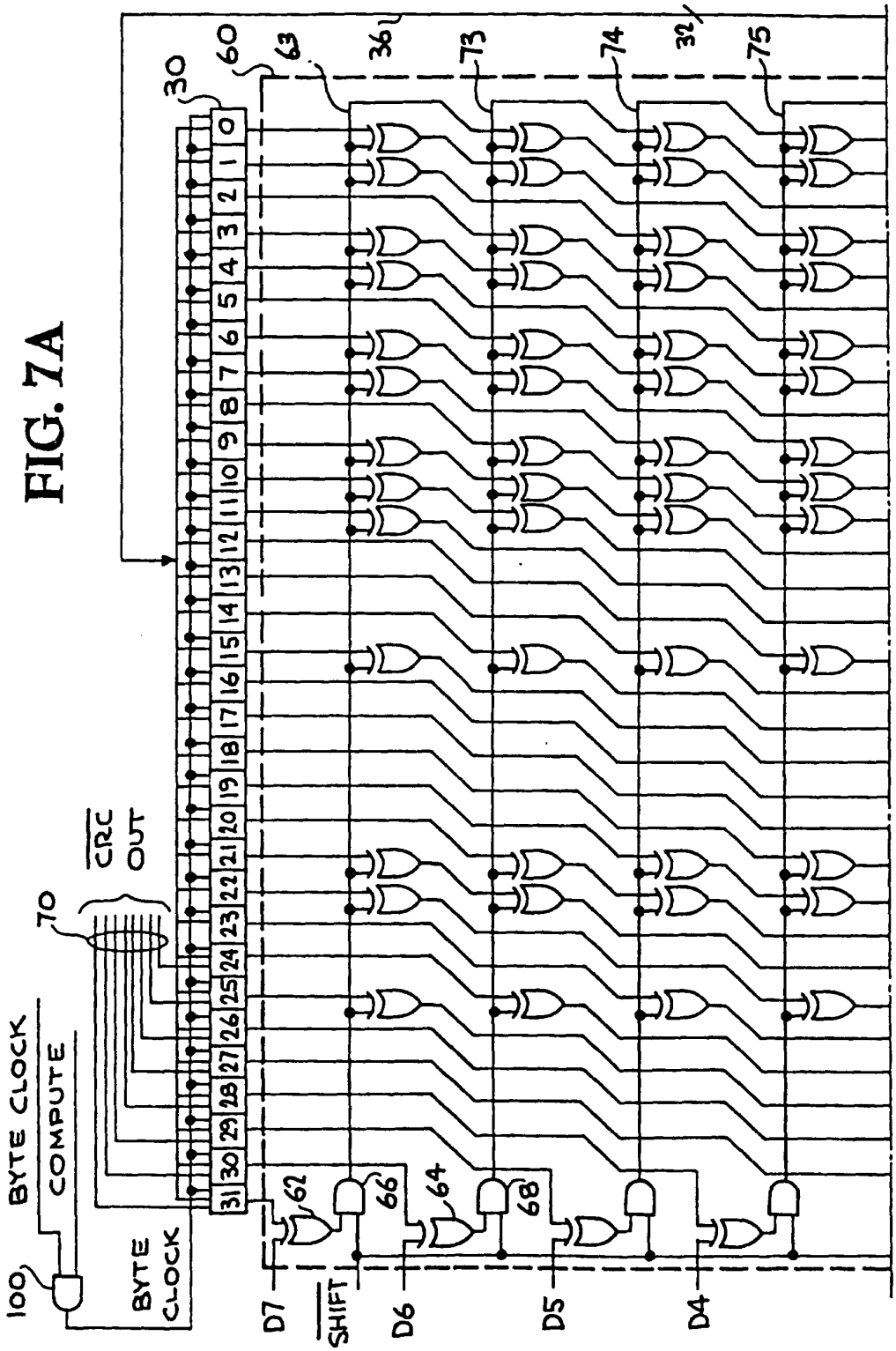

"B" = HOLD SELECT
"C" = COMPUTE SELECT
"A" = PRESET SELECT
"1" PRESET
A
MUX
B
C
86
88
32
30
CLOCK
CRC CHECKSUM REGISTER
Q
32
90
32
91
DATA INPUT
EXCLUSIVE - OR AND SHIFT ARRAY
36
84
32

FIG. 8

FIG. 7A


30
60
63
36
73
74
32
75
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
70
CRC OUT
BYTE CLOCK
COMPUTE
100
BYTE CLOCK
62
66
64
68
D7
SHIFT
D6
D5
D4

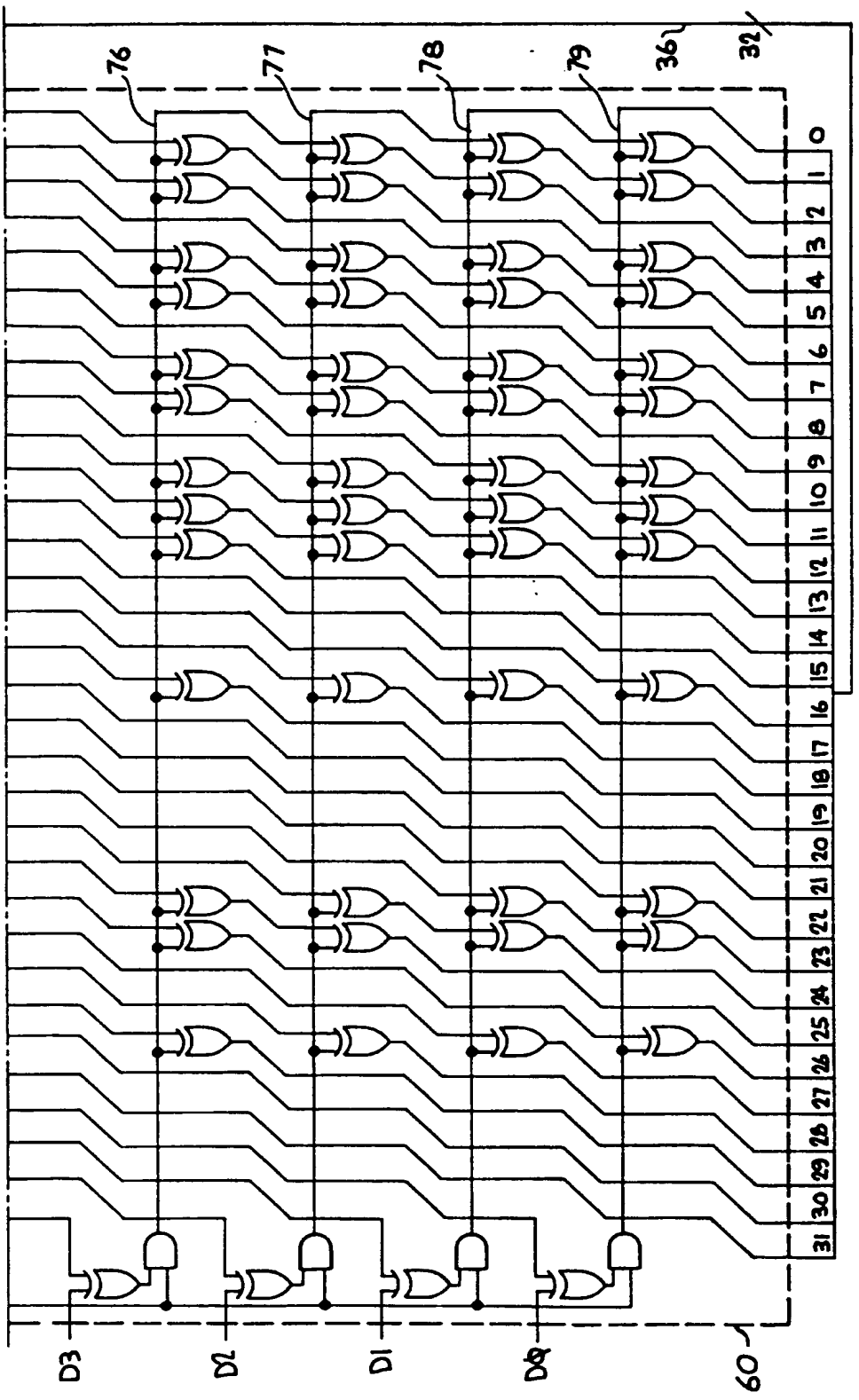

76
77
78
79
36
32
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
D3
D2
D1
D0
60

FIG. 7B

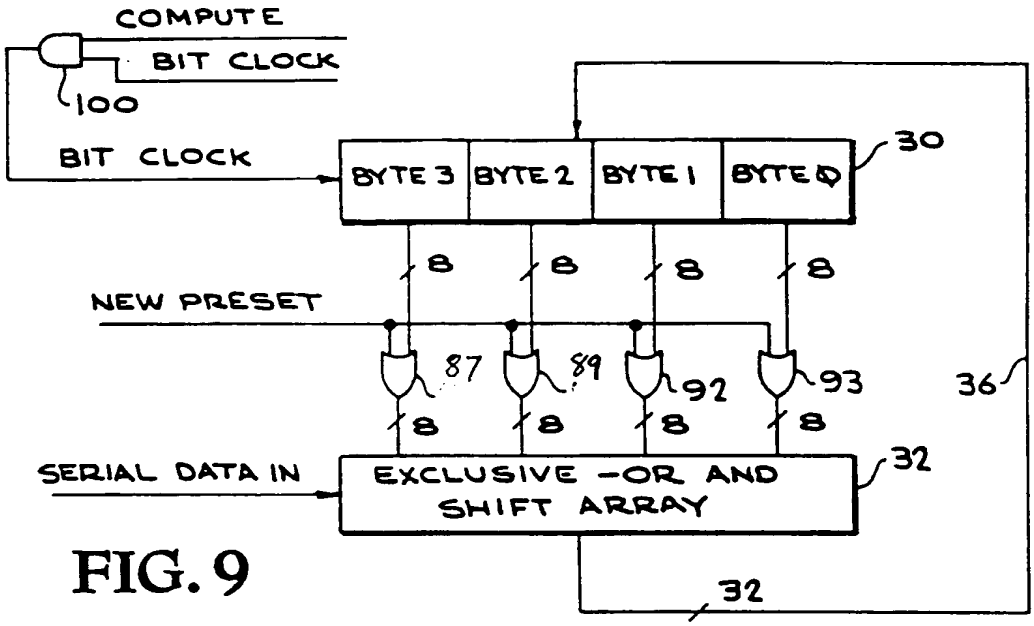

COMPUTE
BIT CLOCK
100
BIT CLOCK
BYTE 3
BYTE 2
BYTE 1
BYTE 0
30
8
8
8
8
NEW PRESET
87
89
92
93
36
8
8
8
8
SERIAL DATA IN
EXCLUSIVE -OR AND SHIFT ARRAY
32
32

FIG. 9

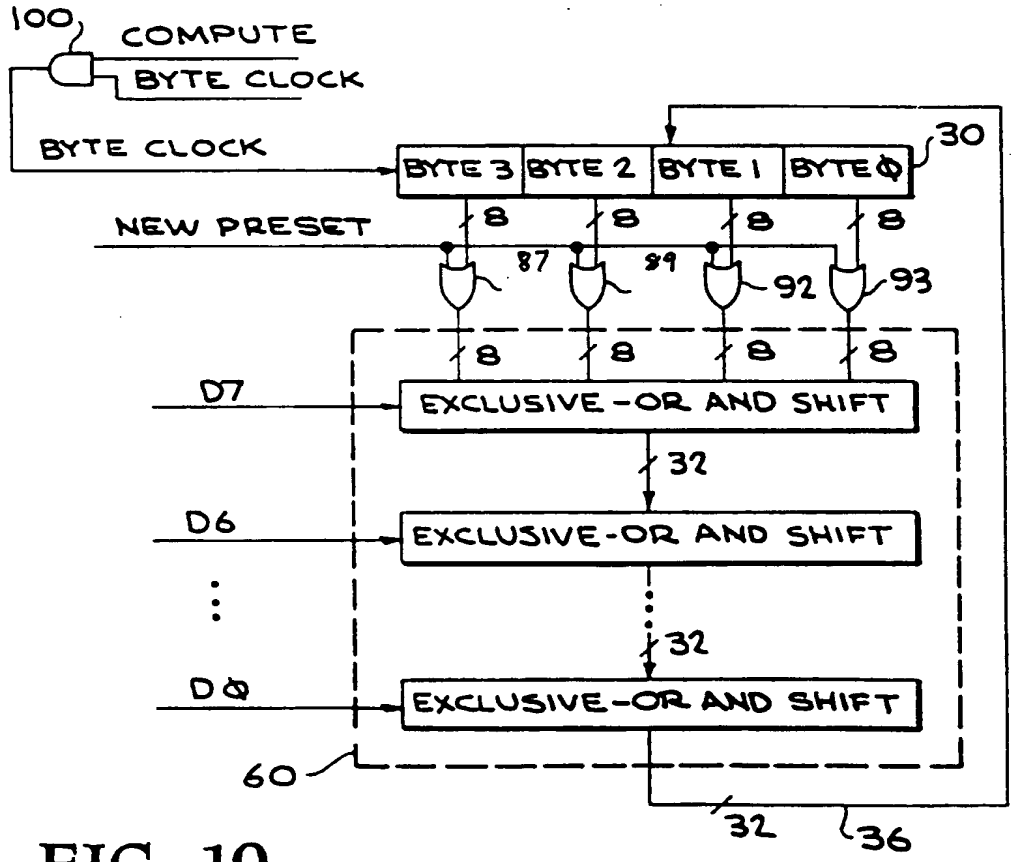

100
COMPUTE
BYTE CLOCK
BYTE CLOCK
BYTE 3
BYTE 2
BYTE 1
BYTE 0
30
NEW PRESET
8
8
8
8
87
89
92
93
8
8
8
8
D7
EXCLUSIVE -OR AND SHIFT
32
D6
EXCLUSIVE-OR AND SHIFT
32
D0
EXCLUSIVE-OR AND SHIFT
60
32
36

FIG. 10

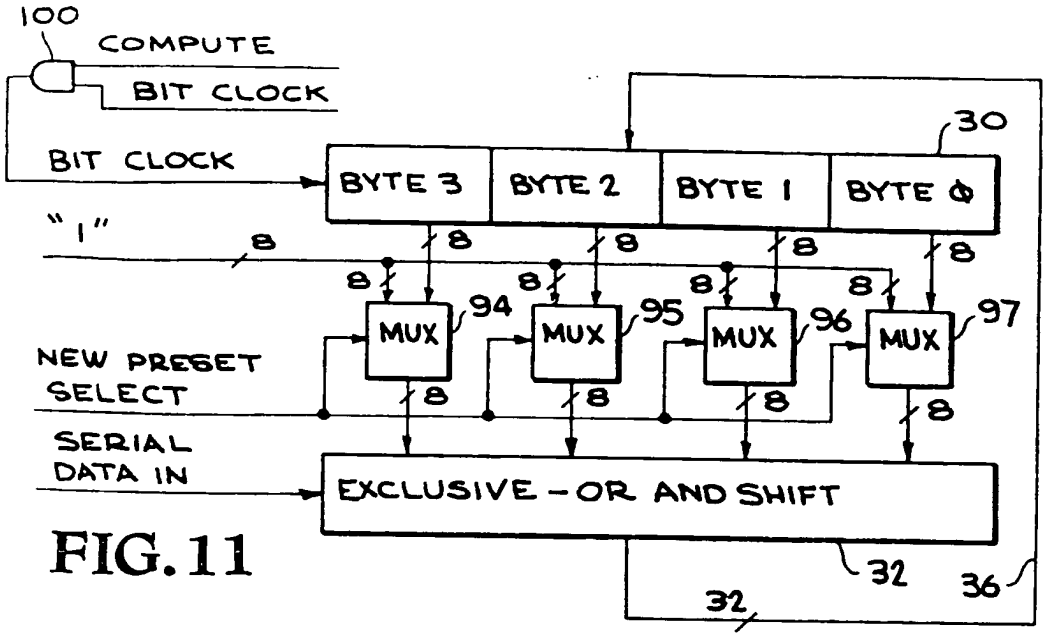

100
COMPUTE
BIT CLOCK
BIT CLOCK
30
BYTE 3
BYTE 2
BYTE 1
BYTE Φ
"1"
8
MUX
94
95
96
97
NEW PRESET SELECT
SERIAL DATA IN
EXCLUSIVE - OR AND SHIFT
32
36

FIG. 11

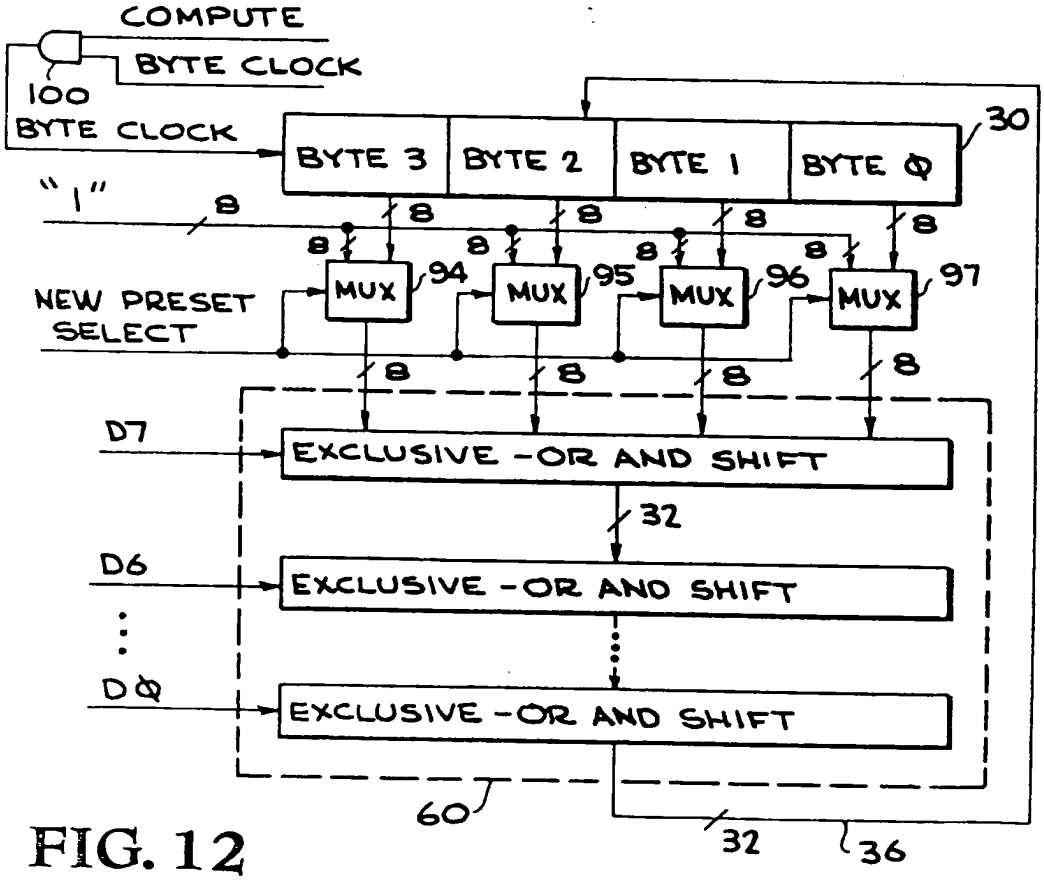

COMPUTE
BYTE CLOCK
100
BYTE CLOCK
30
BYTE 3
BYTE 2
BYTE 1
BYTE Φ
"1"
8
MUX
94
95
96
97
NEW PRESET SELECT
D7
D6
DΦ
EXCLUSIVE - OR AND SHIFT
32
60
36

FIG. 12

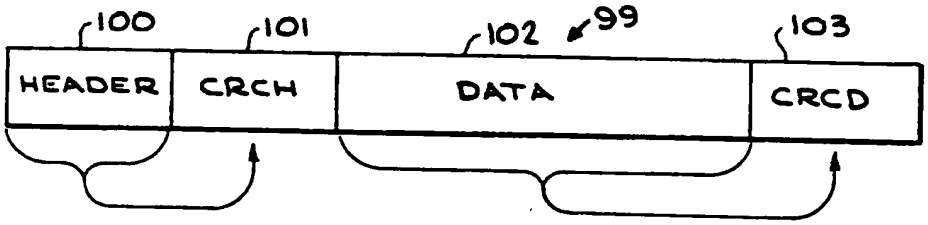


FIG. 14A

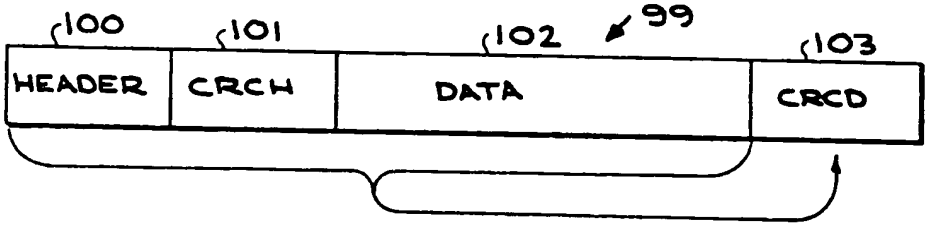


FIG. 14B

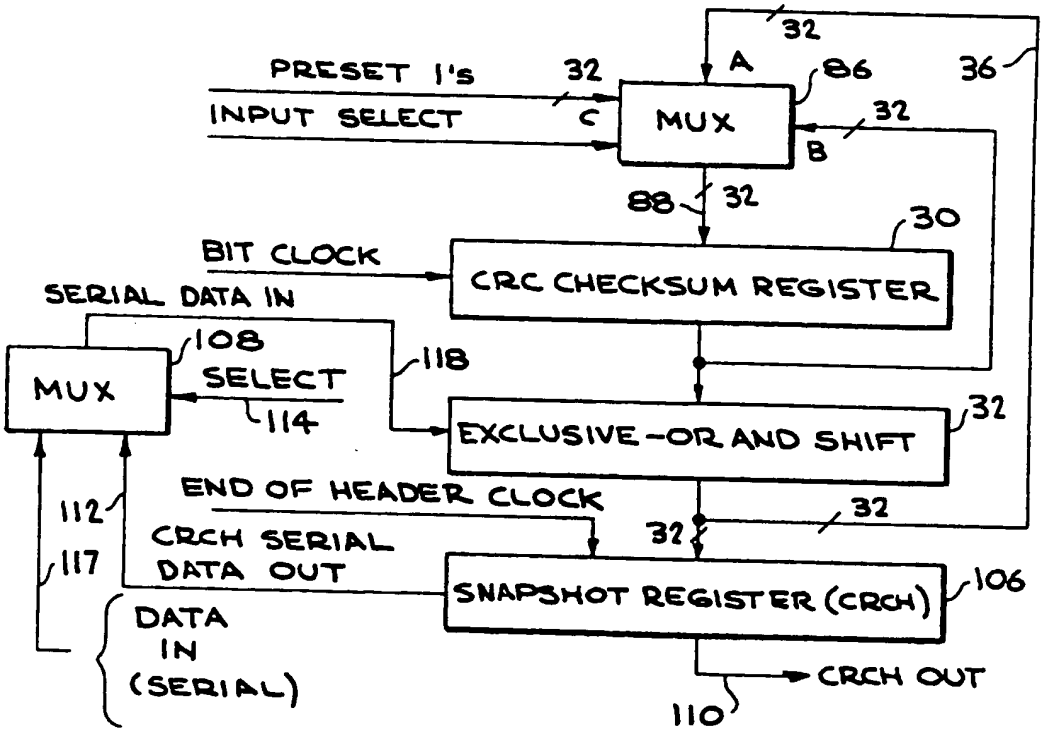


FIG. 15

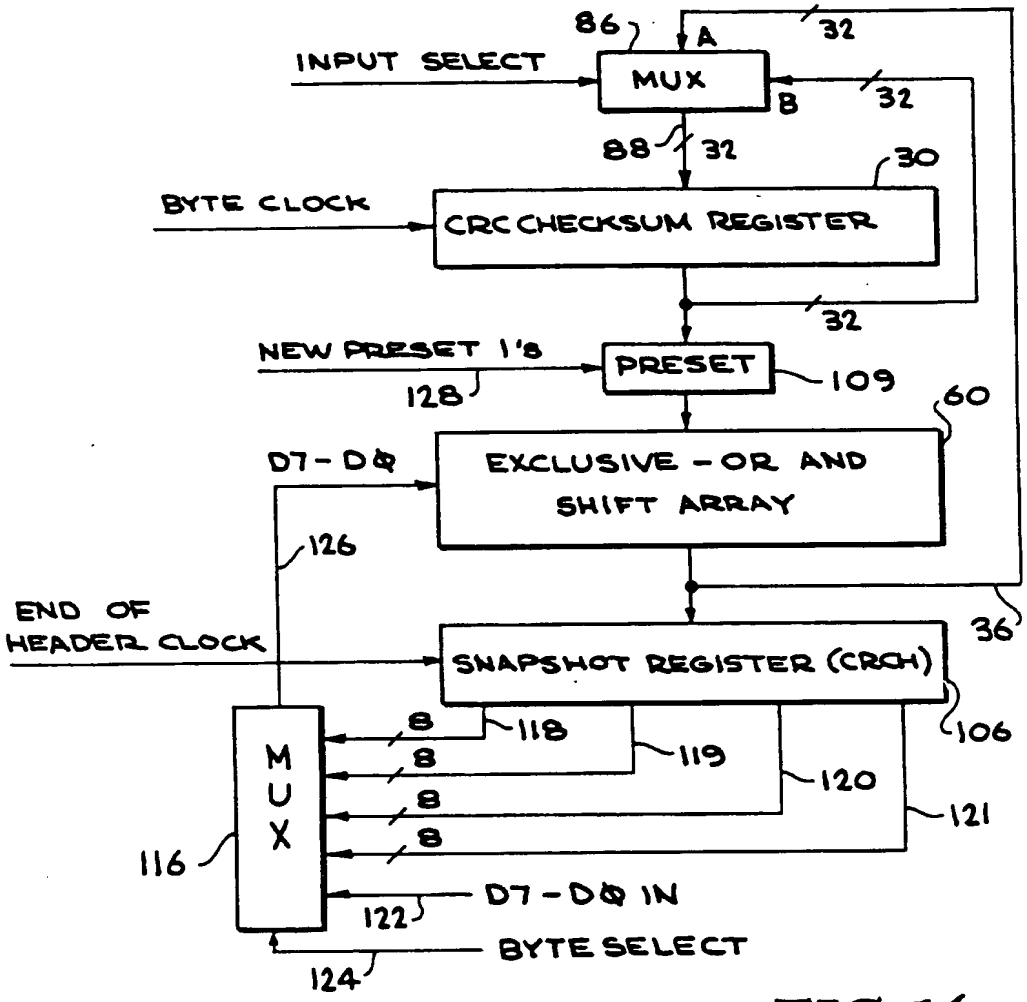

86
A
32
INPUT SELECT
MUX
B
32
88
32
30
BYTE CLOCK
CRC CHECKSUM REGISTER
32
NEW PRESET 1'S
PRESET
109
128
60
D7 - D0
EXCLUSIVE - OR AND
SHIFT ARRAY
126
END OF
HEADER CLOCK
36
SNAPSHOT REGISTER (CRCH)
106
8
118
119
120
121
8
8
8
M
U
X
116
D7 - D0 IN
122
BYTE SELECT
124

FIG. 16

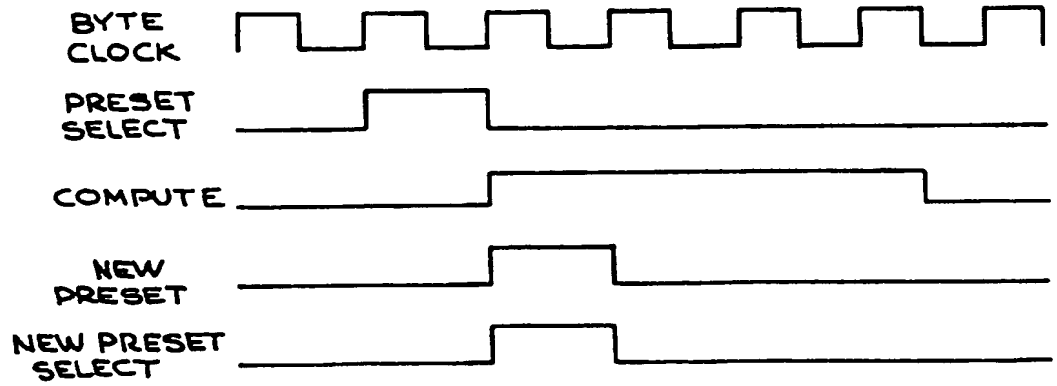

BYTE
CLOCK
PRESET
SELECT
COMPUTE
NEW
PRESET
NEW PRESET
SELECT

FIG. 13

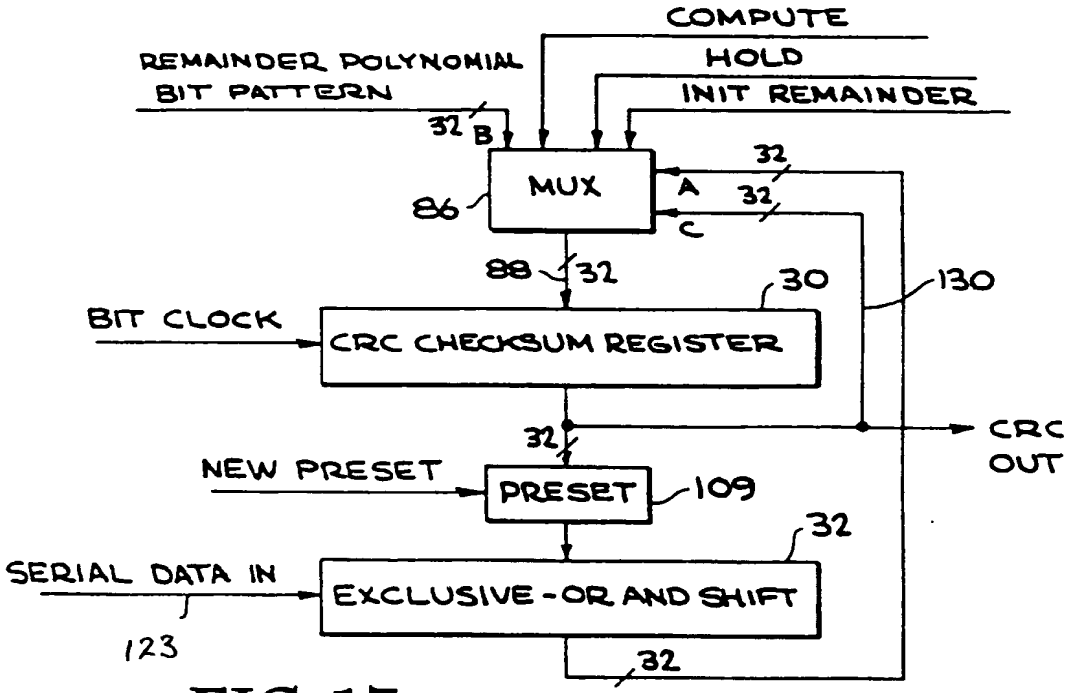

COMPUTE
HOLD
INIT REMAINDER
REMAINDER POLYNOMIAL
BIT PATTERN
32
B
MUX
86
A
C
32
32
88
32
30
130
BIT CLOCK
CRC CHECKSUM REGISTER
CRC
OUT
32
NEW PRESET
PRESET
109
32
SERIAL DATA IN
EXCLUSIVE-OR AND SHIFT
123
32

FIG. 17

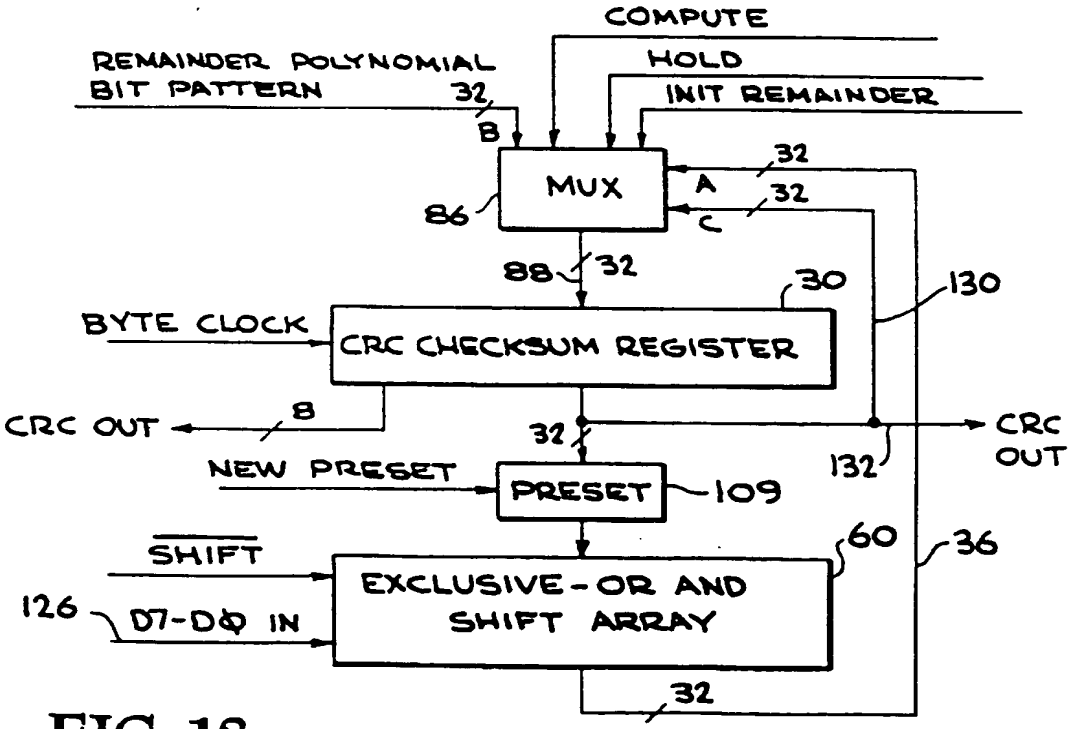

COMPUTE
HOLD
INIT REMAINDER
REMAINDER POLYNOMIAL
BIT PATTERN
32
B
MUX
86
A
C
32
32
88
32
30
130
BYTE CLOCK
CRC CHECKSUM REGISTER
CRC OUT
8
32
CRC
OUT
132
NEW PRESET
PRESET
109
SHIFT
60
36
126
D7-D0 IN
EXCLUSIVE-OR AND
SHIFT ARRAY
32

FIG. 18

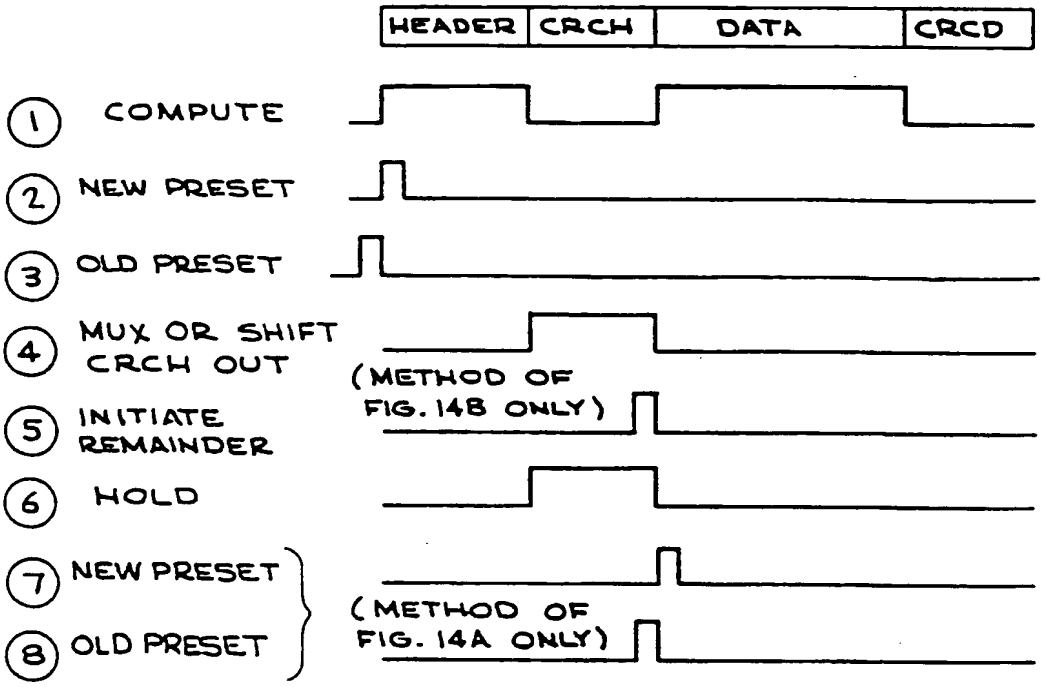

HEADER
CRCH
DATA
CRCD
1 COMPUTE
2 NEW PRESET
3 OLD PRESET
4 MUX OR SHIFT CRCH OUT
(METHOD OF FIG. 14B ONLY)
5 INITIATE REMAINDER
6 HOLD
7 NEW PRESET
8 OLD PRESET
(METHOD OF FIG. 14A ONLY)

FIG. 19

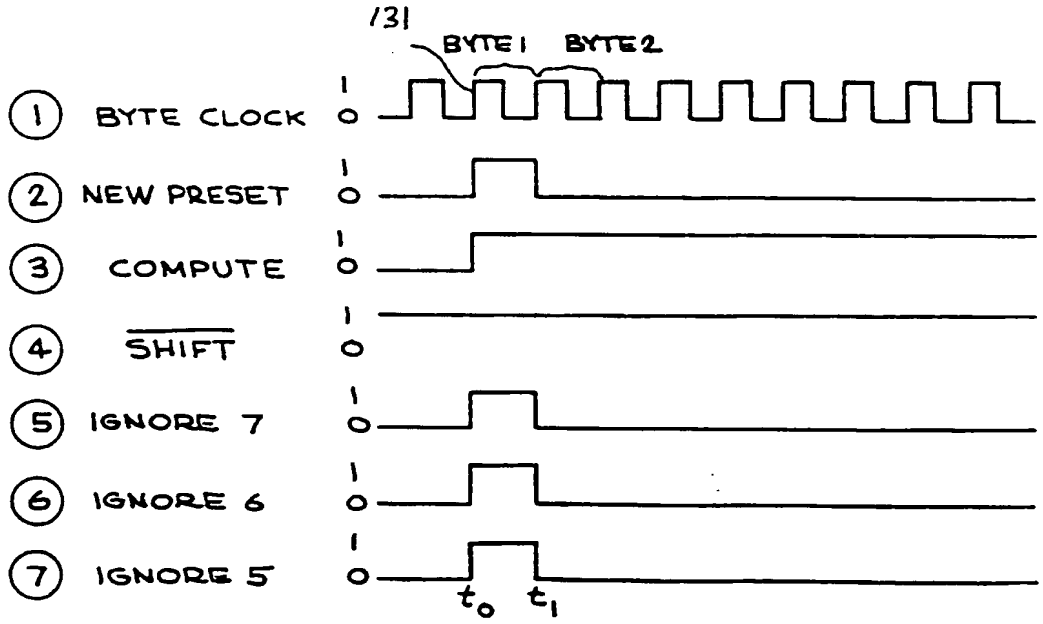
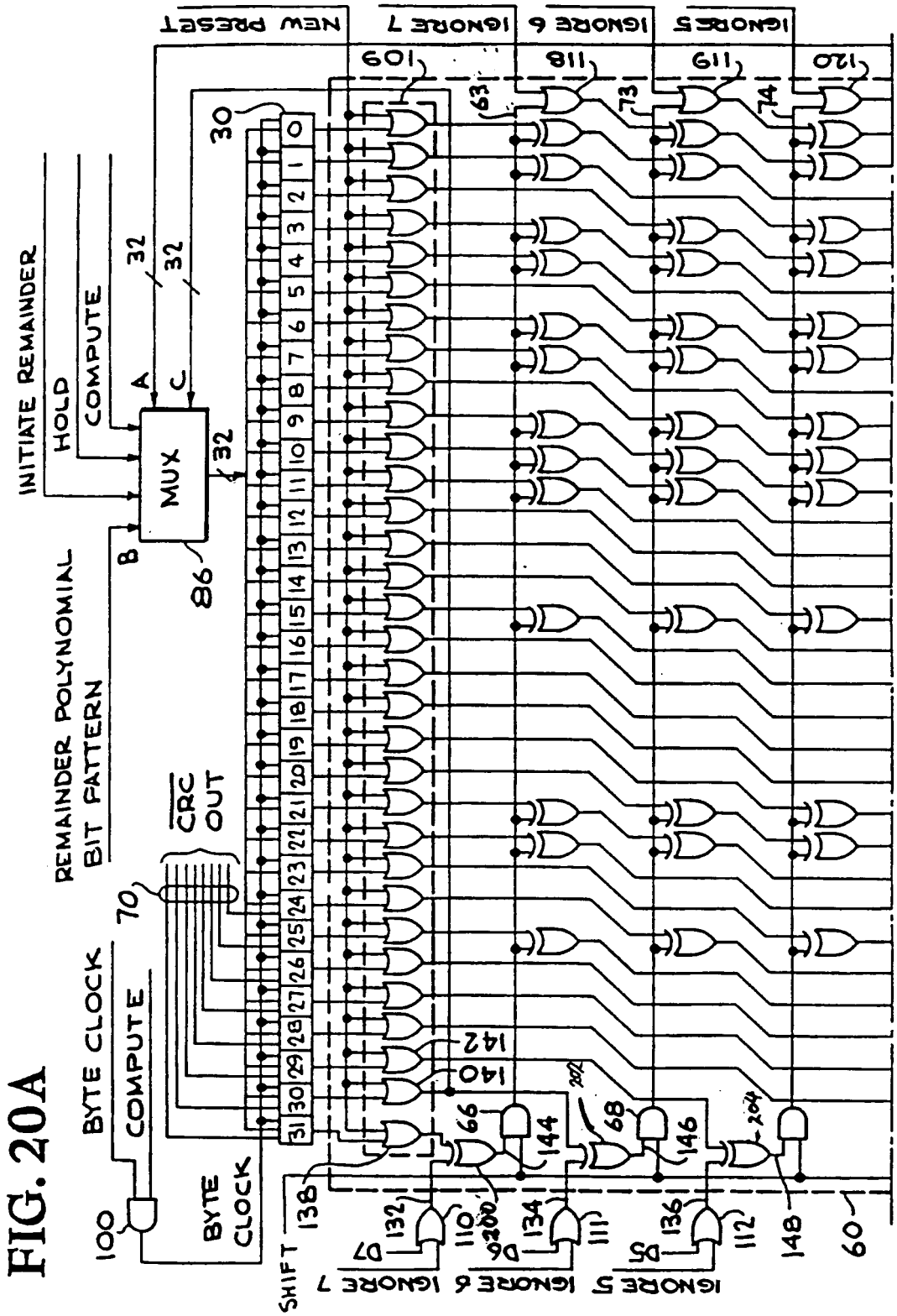

131
BYTE1
BYTE2
1 BYTE CLOCK
2 NEW PRESET
3 COMPUTE
4 SHIFT
5 IGNORE 7
6 IGNORE 6
7 IGNORE 5
1
0
t_0
t_1

FIG. 21

FIG. 20A


INITIATE REMAINDER
HOLD
COMPUTE
REMAINDER POLYNOMIAL
BIT PATTERN
BYTE CLOCK
COMPUTE
MUX
A
B
C
32
32
32
30
86
70
CRC OUT
NEW PRESET
IGNORE 7
IGNORE 6
IGNORE 5
109
118
119
120
63
73
74
100
BYTE CLOCK
SHIFT
138
132
110
134
111
136
112
148
60
66
144
68
146
140
142
202
204
200
D7
D6
D5

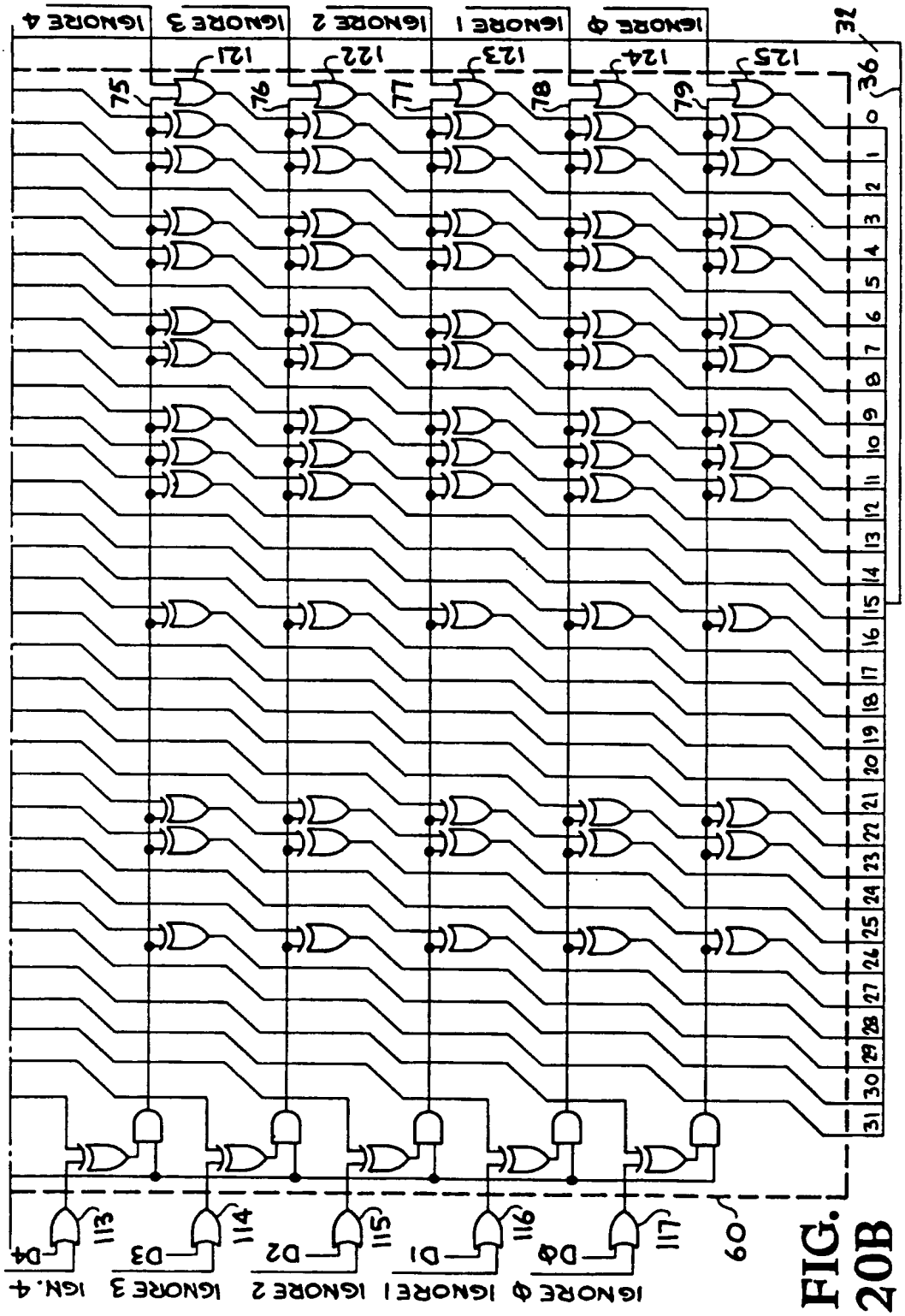

IGNORE 4
IGNORE 3
IGNORE 2
IGNORE 1
IGNORE 0
121
122
123
124
125
75
76
77
78
79
36
31
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
60
113
114
115
116
117
D4
D3
D2
D1
D0
IGN. 4
IGNORE 3
IGNORE 2
IGNORE 1
IGNORE 0

FIG. 20B